



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E  
INDUSTRIAL**

**CARRERA DE TELECOMUNICACIONES**

**Tema:**

---

**SISTEMA ELECTRÓNICO DE REHABILITACIÓN Y MONITOREO PARA  
PERSONAS QUE PADECEN LA ENFERMEDAD DE PARKINSON**

---

Trabajo de Integración Curricular Modalidad: Proyecto de Investigación, presentado  
previo a la obtención del título de Ingeniero en Telecomunicaciones

**ÁREA:** Comunicaciones

**LÍNEA DE INVESTIGACIÓN:** Tecnologías de Comunicación

**AUTOR:** Luis Humberto Cuchipe Diaz

**TUTOR:** Dr. Freddy Geovanny Benalcázar Palacios

**Ambato - Ecuador**

**marzo – 2023**

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del Trabajo de Integración Curricular con el tema: SISTEMA ELECTRÓNICO DE REHABILITACIÓN Y MONITOREO PARA PERSONAS QUE PADECEN LA ENFERMEDAD DE PARKINSON, desarrollado bajo la modalidad Proyecto de Investigación por el señor Luis Humberto Cuchiye Diaz, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 de las segundas reformas al Reglamento para la ejecución de la Unidad de Integración Curricular y la obtención del título de tercer nivel, de grado en la Universidad Técnica de Ambato y sus reformas y el numeral 7.4 del respectivo instructivo del reglamento.

Ambato, marzo 2023.

-----  
Dr. Freddy Geovanny Benalcázar Palacios  
TUTOR

## AUTORÍA

El presente trabajo de Integración Curricular titulado: SISTEMA ELECTRÓNICO DE REHABILITACIÓN Y MONITOREO PARA PERSONAS QUE PADECEN LA ENFERMEDAD DE PARKINSON es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, marzo 2023.



---

Luis Humberto Cuchipec Diaz

C.C. 1501056426

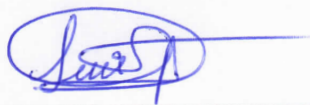
AUTOR

## **DERECHOS DE AUTOR**

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Integración Curricular como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Integración Curricular en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, marzo 2023.



---

Luis Humberto Cuchiye Diaz

C.C. 1501056426

AUTOR

## **APROBACIÓN TRIBUNAL DE GRADO**

En calidad de par calificador del Informe Final del Trabajo de Integración Curricular presentado por el señor Luis Humberto Cuchipe Diaz, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA ELECTRÓNICO DE REHABILITACIÓN Y MONITOREO PARA PERSONAS QUE PADECEN LA ENFERMEDAD DE PARKINSON, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 de las segundas reformas al Reglamento para la ejecución de la Unidad de Integración Curricular y la obtención del título de tercer nivel, de grado en la Universidad Técnica de Ambato y sus reformas y al numeral 7.6 del respectivo instructivo del reglamento. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, marzo 2023.

-----

Ing. Pilar Urrutia, Mg.

PRESIDENTE DEL TRIBUNAL

-----

Ing. Santiago Altamirano, Mg.

PROFESOR CALIFICADOR

-----

Ing. Clara Sánchez, Mg.

PROFESOR CALIFICADOR

## **DEDICATORIA**

*El presente proyecto de titulación se lo dedico a mis padres, Guadalupe y Luis, por ser el pilar fundamental durante toda mi vida, por todo el esfuerzo y sacrificio realizado con el fin de posibilitar todo lo necesario para realizar mis estudios.*

*A mis hermanos por su apoyo incondicional, por sus consejos para afrontar los retos que se me han presentado a lo largo de mi vida.*

*A mi sobrina Samantha que llego a este mundo hace un año trayendo nuevas alegrías para la familia.*

*A mi familia, amigos y compañeros, porque me han brindado su ayuda durante mi etapa universitaria.*

***Lucho***

## **AGRADECIMIENTOS**

*A mis padres y toda mi familia, por su apoyo incondicional para seguir adelante y poder cumplir mis metas.*

*A mis amigos, quienes estuvieron desde el inicio de mi etapa universitaria compartiendo incontables momentos de alegría, sufrimiento y melancolía. Gracias por brindándome su apoyo para seguir adelante y cumplir mi objetivo, de corazón “Ingeniebríos”.*

*A mi tutor, el Dr. Freddy Benalcázar, por su apoyo y asesoramiento durante todo el desarrollo de este proyecto de investigación.*

***Lucho***

## ÍNDICE GENERAL

APROBACIÓN DEL TUTOR.....	II
AUTORÍA.....	III
DERECHOS DE AUTOR.....	IV
APROBACIÓN TRIBUNAL DE GRADO .....	V
DEDICATORIA .....	VI
AGRADECIMIENTOS .....	VII
ÍNDICE GENERAL.....	VIII
ÍNDICE DE TABLAS .....	XII
ÍNDICE DE FIGURAS.....	XIII
RESUMEN EJECUTIVO .....	XVI
ABSTRACT.....	XVII
CAPÍTULO I.....	1
MARCO TEÓRICO.....	1
1.1 Tema de investigación.....	1
1.1.1 Planteamiento del problema.....	1
1.2 Antecedentes Investigativos .....	2
1.3 Fundamentación Teórica .....	5
1.3.1 Enfermedad de Parkinson .....	5
1.3.2 Diagnóstico en la enfermedad de Parkinson .....	6
1.3.3 Síntomas de la Enfermedad de Parkinson.....	7
1.3.4 Tratamiento .....	9
1.3.5 Terapias físicas o fisioterapia.....	9
1.3.6 Terapia ocupacional .....	9
1.3.7 Logopedia.....	10
1.3.8 Sensores.....	11
1.3.9 Almacenamiento en la nube .....	12



1.4	Objetivos.....	12
1.4.1	Objetivo general.....	12
1.4.2	Objetivos específicos .....	12
CAPÍTULO II .....		13
METODOLOGÍA .....		13
2.1	Materiales .....	13
2.2	Métodos .....	13
2.2.1	Modalidad de Investigación .....	13
2.2.2	Recolección de Información .....	14
2.2.3	Procesamiento y Análisis de Datos .....	14
CAPITULO III.....		15
RESULTADOS Y DISCUSIÓN.....		15
3.1	Análisis y discusión de los resultados .....	15
3.1.1	Desarrollo de la propuesta.....	15
3.1.2	ETAPA 1.- Adquisición.....	15
3.1.3	ETAPA 2.- Procesamiento y transmisión .....	18
3.1.4	ETAPA 3.- Servicio en la nube y visualización.....	21
3.1.5	Estancia en la Nube.....	27
3.1.6	Instalación del Bróker Mosquitto.....	32
3.1.7	Instalación de MariaDB .....	33
3.1.8	Instalación del servidor Apache .....	34
3.1.9	Instalación de PHP MyAdmin para administración de la base de datos 35	
3.1.10	Instalación de Node-Red para creación y acceso a la interfaz gráfica.	36
3.1.11	Conexión con MySql.....	37
3.2	Configuración de Node-Red.....	40
3.2.1	Configuración de Node-Red para el ingreso de datos de los pacientes	40

3.2.2	Configuración de Node-Red para almacenar los datos del sensor MPU6050 .....	41
3.2.3	Configuración de Node-Red para almacenar datos del Dispositivo de terapia	43
3.3	Flujograma de procesos para el monitoreo con el sensor MPU6050 .....	47
3.4	Adquisición de datos con el Sensor MPU6050 .....	48
3.4.1	Cálculo de la amplitud de los temblores .....	49
3.4.2	Cálculo de las Frecuencias de los Temblores .....	50
3.5	Diseño esquemático.....	52
3.5.1	Esquema de conexión del dispositivo de monitoreo .....	52
3.5.2	Esquema de conexión del circuito para terapias de pacientes con EP .	52
3.6	Diseño del circuito en PCB .....	53
3.6.1	Dispositivo monitoreo de temblores para EP.....	53
3.6.2	Dispositivo de terapia.....	55
3.7	Diseño del case .....	57
3.8	Dispositivo de terapia convencional.....	60
3.9	Soporte para el dispositivo sensor de temblores para EP .....	61
3.10	Pruebas de funcionamiento.....	62
3.10.1	Vinculación del Dispositivo Electrónico de Monitoreo de Temblores con la red inalámbrica.....	62
3.10.2	Test de evaluación.....	62
3.10.3	Resultados de las evaluaciones .....	65
3.10.6	Resultado del ejercicio de terapia ocupacional .....	75
3.12	Presupuesto .....	81
CAPITULO IV.....		84
CONCLUSIONES Y RECOMENDACIONES.....		84
4.1.	Conclusiones .....	84

4.2. Recomendaciones.....	85
BIBLIOGRAFIA.....	86
ANEXOS.....	90
Anexo 1. Validación Lcda. Liliana Gabriela Sunta Moreno.....	91
Anexo 2. Datasheed Tarjeta ESP8266 .....	92
Anexo 3. Datasheed Sensor MPU6050.....	94
Anexo 4. Dimensiones de la carcasa del sistema electrónico de monitoreo .....	97
Anexo 5. Dimensiones de la carcasa del sistema electrónico de rehabilitación ....	98
Anexo 6. Código de Arduino dispositivo de monitoreo .....	99
Anexo 7. Código de Arduino dispositivo de terapia .....	107

## ÍNDICE DE TABLAS

<b>Tabla 1. Clasificación del temblor [18]</b> .....	8
<b>Tabla 2. Comparación de los sensores inerciales [25]</b> .....	16
<b>Tabla 3. Especificaciones técnicas de la batería [26]</b> .....	17
<b>Tabla 4. Especificaciones técnicas modulo TP4056 [27]</b> .....	18
<b>Tabla 5. Comparación de los microcontroladores [28]</b> .....	19
<b>Tabla 6. Comparación de las tecnologías inalámbricas [29]</b> .....	21
<b>Tabla 7. Comparación de las plataformas Cloud Computing [30]</b> .....	22
<b>Tabla 8. Comparación de los protocolos de comunicación [31]</b> .....	24
<b>Tabla 9. Comparación de las bases de datos [32] [33]</b> .....	25
<b>Tabla 10. Datos de pacientes evaluados</b> .....	65
<b>Tabla 11. Resultado de la evaluación realizada al paciente sano NN1</b> .....	68
<b>Tabla 12. Resultado de la evaluación realizada al paciente sano NN2</b> .....	69
<b>Tabla 13. Resultado de la evaluación realizada al paciente NN3 con EP</b> .....	72
<b>Tabla 14. Resultados de la evaluación realizada al paciente NN4 con EP</b> .....	74
<b>Tabla 15. Tiempo empleado para realizar la terapia ocupacional por los pacientes sanos NN1 y NN2</b> .....	77
<b>Tabla 16. Tiempo empleado para realizar la terapia ocupacional por los pacientes NN3 y NN4 con EP</b> .....	78
<b>Tabla 17. Presupuestos del proyecto</b> .....	83

## ÍNDICE DE FIGURAS

<b>Figura 1. Juego de terapia ocupacional.....</b>	<b>10</b>
<b>Figura 2. Arquitectura del dispositivo de monitoreo de las señales del Parkinson .....</b>	<b>15</b>
<b>Figura 3. Batería Li – Ion.....</b>	<b>17</b>
<b>Figura 4. Modulo 4056 cargador de batería.....</b>	<b>18</b>
<b>Figura 5. Tarjeta ESP8266.....</b>	<b>20</b>
<b>Figura 6. Suscripción de Azure Education.....</b>	<b>27</b>
<b>Figura 7. Creación del grupo de recursos.....</b>	<b>28</b>
<b>Figura 8. Validación del grupo de recursos.....</b>	<b>28</b>
<b>Figura 9. Creación del servidor Ubuntu Server.....</b>	<b>29</b>
<b>Figura 10. Características del servidor web.....</b>	<b>30</b>
<b>Figura 11. Configuración de la autenticación SSH.....</b>	<b>30</b>
<b>Figura 12. Servidor web creado.....</b>	<b>31</b>
<b>Figura 13. Habilitación de puertos en Microsoft Azure.....</b>	<b>32</b>
<b>Figura 14. Bróker de Mosquitto funcionando correctamente.....</b>	<b>33</b>
<b>Figura 15. Servicio de MariaDB funcionando correctamente.....</b>	<b>34</b>
<b>Figura 16. Servidor Apache funcionando correctamente.....</b>	<b>35</b>
<b>Figura 17. phpMyAdmin en el Navegador.....</b>	<b>36</b>
<b>Figura 18. pm2 instalado correctamente y ejecutando Node-Red.....</b>	<b>37</b>
<b>Figura 19. Base de datos.....</b>	<b>37</b>
<b>Figura 20. Almacenamiento de los datos del paciente.....</b>	<b>38</b>
<b>Figura 21. Almacenamiento de los datos del sensor MPU6050.....</b>	<b>38</b>
<b>Figura 22. Almacenamiento de los datos de tiempo de la terapia.....</b>	<b>39</b>
<b>Figura 23. Relación de tablas por claves foráneas.....</b>	<b>39</b>
<b>Figura 24. Interfaz web de Node-Red.....</b>	<b>40</b>
<b>Figura 25. Nodo Form.....</b>	<b>41</b>
<b>Figura 26. Ingreso de datos a la base de datos.....</b>	<b>41</b>
<b>Figura 27. Nodo table.....</b>	<b>41</b>
<b>Figura 28. Nodo funtion.....</b>	<b>42</b>
<b>Figura 29. Nodo mysql.....</b>	<b>42</b>
<b>Figura 30. Nodo mqtt in y Nodo json.....</b>	<b>42</b>
<b>Figura 31. Lectura y almacenamiento de datos.....</b>	<b>43</b>

<b>Figura 32. Representación gráfica de los datos .....</b>	<b>43</b>
<b>Figura 33. Almacenamiento de datos en MariaDB .....</b>	<b>43</b>
<b>Figura 34. Representación de datos en nodos gauge.....</b>	<b>44</b>
<b>Figura 35. Creación de tablas y grupos.....</b>	<b>44</b>
<b>Figura 36. Interfaz del Dashboard para el ingreso de pacientes .....</b>	<b>45</b>
<b>Figura 37. Datos del paciente en la Interfaz Monitoreo Movimiento.....</b>	<b>45</b>
<b>Figura 38. Interfaz gráfica Monitoreo de Movimiento.....</b>	<b>46</b>
<b>Figura 39. Interfaz gráfica del dashboard Terapia Juego.....</b>	<b>46</b>
<b>Figura 40. Diagrama de flujo monitoreo sensor MPU6050.....</b>	<b>48</b>
<b>Figura 41. Librerías de Arduino para el sensor MPU6050.....</b>	<b>48</b>
<b>Figura 42. Cálculo de aceleraciones y amplitudes.....</b>	<b>50</b>
<b>Figura 43. Variables para el cálculo de la FFT .....</b>	<b>50</b>
<b>Figura 44. Código del cálculo de la frecuencia en Arduino.....</b>	<b>51</b>
<b>Figura 45. Librerías para la transmisión de datos.....</b>	<b>51</b>
<b>Figura 46. Esquema de conexión de los componentes electrónicos del dispositivo sensor de monitoreo de temblores para EP .....</b>	<b>52</b>
<b>Figura 47. Esquema de conexión de los componentes electrónicos del dispositivo de terapia.....</b>	<b>53</b>
<b>Figura 48. Diseño del PCB de monitoreo .....</b>	<b>54</b>
<b>Figura 49. Diagrama de pistas de la PCB diseñada .....</b>	<b>54</b>
<b>Figura 50. Implementación del circuito final .....</b>	<b>55</b>
<b>Figura 51. Diseño del PCB de dispositivo de terapia .....</b>	<b>55</b>
<b>Figura 52. Diagrama de pistas de la PCB diseñada .....</b>	<b>56</b>
<b>Figura 53. Implementación del dispositivo de terapia.....</b>	<b>56</b>
<b>Figura 54. Case en 3D para placa de monitoreo .....</b>	<b>57</b>
<b>Figura 55. Case en 3D placa del dispositivo de terapia .....</b>	<b>58</b>
<b>Figura 56. Case para insertar los objetos esféricos.....</b>	<b>58</b>
<b>Figura 57. Colocación de la placa de monitoreo en el case.....</b>	<b>59</b>
<b>Figura 58. Colocación de la placa de terapias en el case .....</b>	<b>59</b>
<b>Figura 59. Cajas contenedoras de objetos circulares para la terapia .....</b>	<b>60</b>
<b>Figura 60. Dispositivo para actividad de terapia convencional .....</b>	<b>60</b>
<b>Figura 61. Soporte para el dispositivo de monitoreo .....</b>	<b>61</b>

<b>Figura 62. Dispositivo de monitoreo de temblores para EP en el soporte para mano .....</b>	<b>61</b>
<b>Figura 63. Conexión del sistema de monitoreo a la red local.....</b>	<b>62</b>
<b>Figura 64. Postura para evaluar temblor postural de las manos .....</b>	<b>63</b>
<b>Figura 65. Ejercicio para la evaluación de Temblor de Acción de las Manos. El ejercicio comienza con la postura (a), después lentamente se pasa a la postura (b) y culmina al adoptar nuevamente la postura (a).....</b>	<b>64</b>
<b>Figura 66. Postura para evaluación Amplitud Temblor en Reposo .....</b>	<b>65</b>
<b>Figura 67. Pruebas de funcionamiento en Pacientes Sanos y con EP .....</b>	<b>66</b>
<b>Figura 68. Resultado del TEST 1 realizado al paciente sano NN1 .....</b>	<b>67</b>
<b>Figura 69. Resultado del TEST 2 realizado al paciente sano NN1 .....</b>	<b>67</b>
<b>Figura 70. Resultado del TEST 3 realizado al paciente sano NN1 .....</b>	<b>68</b>
<b>Figura 71. Resultados del TEST 1 del paciente NN3 con EP.....</b>	<b>70</b>
<b>Figura 72. Resultados del TEST 2 del paciente NN3 con EP.....</b>	<b>71</b>
<b>Figura 73. Resultados del TEST 3 del paciente NN3 con EP.....</b>	<b>71</b>
<b>Figura 74. Resultados del TEST 1 del paciente NN4 con EP.....</b>	<b>73</b>
<b>Figura 75. Resultados del TEST 2 del paciente NN4 con EP.....</b>	<b>73</b>
<b>Figura 76. Resultados del TEST 3 del paciente NN4 con EP.....</b>	<b>74</b>
<b>Figura 77. Ejercicio de terapia ocupacional .....</b>	<b>76</b>
<b>Figura 78. Gráfica evolutiva en la terapia convencional de motricidad fina de la mano derecha del paciente NN3.....</b>	<b>79</b>
<b>Figura 79. Gráfica evolutiva en la terapia convencional de motricidad fina de la mano izquierda del paciente NN3.....</b>	<b>79</b>
<b>Figura 80. Gráfica evolutiva en la terapia convencional de motricidad fina de la mano derecha del paciente NN4.....</b>	<b>80</b>
<b>Figura 81. Gráfica evolutiva en la terapia convencional de motricidad fina de la mano izquierda del paciente NN4.....</b>	<b>80</b>

## RESUMEN EJECUTIVO

El Parkinson es un trastorno neurodegenerativo crónico presente en un gran porcentaje de la población, que afecta a la calidad de vida de quienes lo padecen. Los profesionales de la salud encargados de tratar dicha enfermedad deben realizar seguimientos de los avances en el tratamiento de sus pacientes. Actualmente, gracias a la tecnología se puede monitorear y llevar un control en entornos digitales.

En el presente proyecto se desarrolló un dispositivo utilizando softwares de código abierto. Para el monitoreo utiliza una tarjeta ESP8266 y el sensor MPU6050 para obtener la aceleración en las diferentes orientaciones y en base a estos datos calcular la amplitud y frecuencia de los movimientos involuntarios de los pacientes, información que es enviada a una base de datos para su almacenamiento y puedan ser visualizados gráficamente para una mejor interpretación por parte del médico tratante, lo que le permitirá tomar decisiones en cuanto a tratamiento y proceso de rehabilitación. Como aporte al proceso de rehabilitación, el sistema electrónico diseñado cuenta una tarjeta ESP8266 e interruptores de posición para controlar la ubicación de esferas metálicas que son transportadas por el paciente para procesos de fortalecimientos de músculos y manejo de motricidad fina.

Tomando en cuenta los criterios de rehabilitación aplicados en terapia convencional de motricidad fina y la clasificación de los temblores que presenta un paciente con Enfermedad de Parkinson, los resultados obtenidos en cada una de las pruebas realizadas con el sistema electrónico desarrollado demuestran la fiabilidad del dispositivo de monitoreo para el seguimiento del paciente y aporta significativamente a procesos de rehabilitación mediante ejercicios que contribuyen al fortalecimiento de la motricidad fina.

**Palabras clave:** Enfermedad de Parkinson, sistema de monitoreo, rehabilitación, terapia convencional.



## ABSTRACT

Parkinson is a chronic neurodegenerative disorder present in a large percentage of the population, which affects the quality of life of those who suffer from it. Health professionals in charge of treating Parkinson's disease must monitor the progress of their patients' treatment. Currently, thanks to technology it is possible to monitor and keep track in digital environments.

In the present project a device was developed using open-source software. For monitoring uses an ESP8266 card and the MPU6050 sensor to obtain the acceleration in different orientations and based on these data calculate the amplitude and frequency of involuntary movements of patients, this information is sent to a database for storage and can be displayed graphically for better interpretation by the treating physician, which will allow him to make decisions regarding treatment and rehabilitation process. As a contribution to the rehabilitation process, the electronic system designed has an ESP8266 card and position switches to control the location of metallic spheres that are transported by the patient for muscle strengthening and fine motor management processes.

Considering the rehabilitation standards applied in conventional fine motor therapy and the classification of tremors presented by a patient with Parkinson's disease, the results obtained in each of the tests performed with the electronic system developed demonstrate the reliability of the monitoring device for patient tracking and contribute significantly to rehabilitation processes through exercises that contribute to the strengthening of fine motor skills.

**Keywords:** Parkinson's disease, monitoring system, rehabilitation, conventional therapy.

# CAPÍTULO I

## MARCO TEÓRICO

### 1.1 Tema de investigación

Sistema electrónico de rehabilitación y monitoreo para personas que padecen la enfermedad de Parkinson.

#### 1.1.1 Planteamiento del problema

Según la Organización mundial de la Salud (OMS) en la publicación trastornos Neurológicos [1] define a la enfermedad del Parkinson (EP) como un trastorno neurodegenerativo crónico, caracterizado por la presencia de síntomas motores y no motores, relacionados con el daño de múltiples estructuras del sistema nervioso central y periférico, se caracteriza por temblor, rigidez, lentitud en los movimientos e inestabilidad postural. En la actualidad uno de los problemas que enfrenta la población es la Enfermedad de Parkinson, considerada como la segunda enfermedad degenerativa más prevalente después del Alzheimer [2]. La EP afecta a 1 de cada 100 mayores de 60 años, también prevé que para 2030 a nivel mundial habrá 12 millones de pacientes de la Enfermedad de Parkinson [3].

En América Latina, [4] en un estudio realizado por la Universidad Internacional de Valencia, menciona que Chile es el país que ha visto aumentada su prevalencia en más de un 20%, por lo que se sitúa en la cabeza de la lista de países de Latinoamérica con EP. Seguidamente, entre un 18 y 19% se encuentran Paraguay, El Salvador, Honduras y Guatemala. Por su parte los países Haití, Panamá, Bolivia, Venezuela, Ecuador, Perú, Puerto Rico y Uruguay han mostrado un aumento entre el 10 y 15% en la prevalencia de la enfermedad de Parkinson. Sin embargo, en Ecuador existe muy poca financiación para el tratamiento de la EP, esto causa que los pacientes no sean atendidos de una manera adecuada y no tengan acceso a terapias que controlen de una manera adecuada la evolución de los síntomas de la EP.

En Ecuador esta problemática va creciendo año por año ya que según datos del estudio realizado sobre la prevalencia de la enfermedad de Parkinson en la Provincia de Manabí, desarrollada por Montalvo J, Albear E, Intriago R, Moreira V y Montalvo S

[5] mencionan que en Ecuador se obtuvo un estimado de 243 casos positivos para enfermedad de Parkinson por cada 100.000 habitantes, con predominio en mayores de 61 años y con una prevalencia de afectación mayor al género masculino en comparación del femenino.

En la actualidad la provincia de Tungurahua no cuenta con campañas que ayuden a la prevención del Parkinson, la última fue hace 4 años en donde el Patronato Provincial de Tungurahua realizó ejercicios mentales a 160 adultos mayores de diferentes sectores de la provincia a través de la campaña de prevención de Parkinson y Alzheimer. Además, se realizó una serie de ejercicios dirigidos a detectar y valorar de forma rápida la presencia de síntomas, rasgos y disfuncionalidades en los procesos cognitivos afectados en la enfermedad de Parkinson. De acuerdo con el informe médico del Centro del Patronato Provincial de Tungurahua; 25 casos fueron diagnosticados con Parkinson siendo más frecuente en hombres que en mujeres. [6]

Actualmente la EP no tiene cura, pero si se puede mantener bajo control de forma muy eficaz. Para tratar dicha enfermedad se utiliza medicamentos basados en dopamina (levodopa) que ayudan a mejorar todos los síntomas motores cardinales. También existe las fisioterapias o ejercicio físico que ayudan al tratamiento de los síntomas motores, principalmente el equilibrio y la postura [7]. Por lo antes expuesto, se plantea realizar el presente trabajo, el cual está enfocado en desarrollar un sistema electrónico de bajo costo y accesible, el cual ayude a monitorear los signos vitales y el temblor parkinsoniano en extremidades superiores conforme el paciente realice un ejercicio de rehabilitación.

## **1.2 Antecedentes Investigativos**

Como punto de partida para el desarrollo del presente proyecto se cuenta con trabajos previos realizados en los últimos años, los cuales están directamente relacionados con el tema, los mismos que se presentan a continuación:

Fabiola Chacha (2017) en la Universidad Politécnica Salesiana sede Cuenca, presentó el trabajo titulado “Diseño, desarrollo e implementación de un sistema inalámbrico para estimular ejercicios de rehabilitación en pacientes con Parkinson”, el sistema está enfocado en la rehabilitación de EP mediante el uso de videojuegos involucrados con

el movimiento del jugador. El videojuego está diseñado con Unity y Arduino, el software Unity permite exportar el videojuego a la plataforma Android. El sistema adquiere los datos del paciente inalámbricamente usando un módulo Bluetooth y un acelerómetro, dichos datos intervienen en el juego y son enviados de manera inalámbrica al dispositivo móvil (Tableta o celular). Mediante pruebas de validación del sistema se obtuvo como resultado la recuperación efectiva de los pacientes en comparación con las terapias convencionales y en adultos mayores fue de gran ayuda para la rehabilitación ya que presentaron mejoras en la motricidad y movilidad de brazos, codos y manos [8].

Bertha Punin y Boris Barzallo (2017) de la Universidad Politécnica Salesiana sede Cuenca, desarrollaron el proyecto titulado “Diseño, desarrollo e implementación de un dispositivo inalámbrico para detección de episodios de congelamiento de la marcha en pacientes con la enfermedad de Parkinson”, el dispositivo inalámbrico está basado en hardware libre para la adquisición en tiempo real de datos de pacientes con la Enfermedad de Parkinson, para luego ser transmitidos a un dispositivo móvil mediante la tecnología Bluetooth, además, cuenta con una aplicación móvil donde se pueden visualizar mediante gráficos los datos de la aceleración en tres ejes (AccX, AccY, AccZ). El dispositivo detecta episodios de congelamiento de la marcha en pacientes con la Enfermedad de Parkinson y mediante un motor vibratorio (ubicado en la intersección del nervio tibial posterior y nervio plantar lateral) se puede activar y desactivar manualmente los estímulos vibratorios para el desbloqueo de la congelación [9].

Franklin Fajardo (2018) en el Departamento de Ciencias de la Computación y Electrónica de la Universidad de la Costa, implementó un “Dispositivo electrónico para la adquisición de señales asociadas a la Enfermedad de Parkinson (EP)”, el cual está realizado mediante tres herramientas de simulación: Proteus, Arduino y LabVIEW. El dispositivo toma los datos del pulso cardiaco, la temperatura y la velocidad de movimiento de las extremidades. Dichos datos son almacenados y analizados mediante una interfaz gráfica en Labview. Como resultado de las pruebas de funcionamiento se pudo discriminar con rigor los sujetos sanos de los sujetos con EP mediante el análisis de datos relacionados con el pulso y la frecuencia cardiaca. [10].

Vicky Lechón (2019) de la Universidad Técnica del Norte, en Ecuador, diseñó un “Dispositivo para medir el temblor en personas con la enfermedad de Parkinson”, el cual consiste en medir los movimientos involuntarios del paciente mediante un acelerómetro MPU- 6050 conectado a una placa electrónica de adquisición de datos, la información obtenida es procesada y mostrada mediante una interfaz gráfica de Matlab. Se realizaron pruebas de funcionamiento del dispositivo con personas sanas y los valores de amplitud y frecuencia permanecieron en cero mientras que en personas con la EP la frecuencia varía de 4 Hz a 6 Hz debido a que los pacientes presentan movimientos involuntarios en sus extremidades superiores [11].

Guillermo Trinidad (2020) de la Universidad de la República, en Uruguay, realizó un “Dispositivo de asistencia ante el congelamiento de la marcha en pacientes con Parkinson” donde desarrolló un dispositivo de señalización capaz de guiar a los pacientes hacia destinos de su elección de trayectorias que toman en cuenta los obstáculos presentes en el camino y los cambios de dirección necesarios. Utiliza técnicas de mapeo y localización mediante SLAM visual y RTAB-Map, que consiste en la construcción de un mapa del entorno, para ello se debe recorrer el edificio de una punta a la otra, ida y vuelta, una única vez. Una vez que se cuenta con el mapa del entorno, se prueba la capacidad del sistema para localizar al usuario en el mismo, para ubicarse en el mapa se lo realiza mediante RTAB-Map. Como resultados se tiene la implementación de una interfaz simple que presenta estímulos capaces de guiar un usuario sin la necesidad de que este desvíe su atención [12].

Francisco Sigcha (2021) de la Universidad Politécnica de Madrid, en España, desarrolla una metodología de “Detección automática de síntomas motores asociados a la Enfermedad de Parkinson mediante relojes y teléfonos móviles inteligentes aplicando técnicas de Inteligencia Artificial”, el cual emplea tecnologías de salud móvil (*m-health*) como relojes y teléfonos inteligentes, dispositivos que se utilizaron para la recolección de datos a partir de una serie de ejercicios que son realizados por un grupo de pacientes con la Enfermedad de Parkinson (EP). Los datos adquiridos permiten extraer información relevante para el seguimiento de los síntomas motores en la EP mediante la aplicación de diversas técnicas de Inteligencia Artificial. La metodología propuesta, basada en la realización de ejercicios estandarizados, muestra una alta capacidad para la implementación de detectores automáticos que emplean

técnicas de aprendizaje automático y profundo para la detección de síntomas específicos, como los temblores de reposo y la detección de la bradicinesia [13].

Carlos Godínez & Luzmila Pro (2022) de la Universidad Nacional de San Marcos, en Perú, realizaron “Una aplicación móvil con técnicas de Machine Learning a partir de datos de sensores inerciales para el monitoreo e identificación de ocurrencias del temblor de Parkinson”, la aplicación móvil esta desarrollada mediante el lenguaje de programación Java y framework Ionic. Mediante la App se puede medir la vibración de la mano, el congelamiento de la marcha e identificar o predecir los valores en los próximos segundos del temblor de Parkinson, dicha aplicación utiliza técnicas lineal y datos de sensores inerciales (acelerómetro y giroscopio) presentes en los teléfonos inteligentes. Además, en la App se puede registrar al usuario, programar la dosificación de medicamentos, registrar los síntomas y ver el historial del paciente. Como resultado se tiene una aplicación móvil con una interfaz gráfica donde se pueden visualizar los resultados de las señales del acelerómetro en Hz [14].

### **1.3 Fundamentación Teórica**

#### **1.3.1 Enfermedad de Parkinson**

La Enfermedad de Parkinson (EP), descrita por el médico inglés Dr. James Parkinson en el año 1817 [11], es el segundo trastorno neurodegenerativo más común, después de la enfermedad de Alzheimer [15]. La EP se caracteriza por la presencia de síntomas cardinales, entre los cuales están: temblor, bradicinesia/acinesia, rigidez muscular, alteración de los reflejos posturales y el trastorno del congelamiento de marcha. También pueden aparecer síntomas relacionados con la salud mental como problemas de memoria o demencia conocidos como síntomas no-motores [13].

Desde un análisis fisiopatológico, la EP es conocida y se distingue por la pérdida progresiva de neuronas pigmentadas en ciertas zonas del sistema nervioso central, siendo la zona más afectada la región de la sustancia negra. A medida que las neuronas pigmentadas desaparecen, se deja de producir una sustancia llamada dopamina, la falta de dopamina hace que las células que controlan el movimiento no puedan enviar mensajes apropiados a los músculos provocando descoordinación en los movimientos.

La EP suele comenzar entre los 55-65 años, pero también se da en personas más jóvenes, de 20 a 55 años [16]. Antes de los 20 años es considerada como enfermedad de Parkinson de inicio juvenil, antes de 50 años es considerada como EP de inicio temprano y luego de 50 años se considera como inicio tardío.

### **1.3.2 Diagnóstico en la enfermedad de Parkinson**

Diagnosticar la enfermedad de Parkinson puede requerir algún tiempo. Los médicos pueden sugerir citas de revisión con neurólogos especializados en trastornos del movimiento para evaluar el padecimiento y sus síntomas [17]. El diagnóstico de la enfermedad de Parkinson es clínico, se basa en reconocer la manifestación de los principales síntomas motores, temblor de reposo, rigidez, inestabilidad postural y bradicinesia, durante las primeras etapas de la enfermedad. Otro criterio que permite diagnosticar la EP es mediante la administración del medicamento levodopa durante un mes, en donde el paciente presenta una respuesta inmunológica con un grado de mejoría del 30 % en la Escala Unificada de Valoración de la Enfermedad de Parkinson (UPDRS) [8].

#### **UPDRS (Escala Unificada de Valoración de la Enfermedad de Parkinson)**

La UPDRS tiene valoraciones múltiples que miden el funcionamiento mental, la conducta y el ánimo; las actividades de la vida cotidiana y la función motora. Se utiliza para evaluar el progreso y el efecto de los tratamientos en las pacientes con EP. Se divide en subescalas del I al IV [18].

El temblor se encuentra en Subescala III correspondiente a exploración motora y sus niveles son:

Temblor de reposo

0 = Ausente.

1 = Leve o presente infrecuentemente.

2 = amplitud moderada pero presente intermitentemente.

3 = Moderado en amplitud y presente la mayor parte del tiempo.

4 = Gran amplitud y presente la mayor parte del tiempo.

Temblor postural

0 = Ausente.

- 1 = Leve; presente en movimientos intencionados.
- 2 = Amplitud moderada; presente en movimientos de intención.
- 3 = Amplitud moderada tanto postural como de intención.
- 4 = Amplitud macada; interfiere con la alimentación.

#### Temblor cinético

- 0 = Ausente.
- 1 = Leve; presente en movimientos intencionados.
- 2 = Amplitud moderada; presente en movimientos de intención.
- 3 = Amplitud moderada tanto postural como de intención.
- 4 = Amplitud macada; interfiere con la alimentación.

### **1.3.3 Síntomas de la Enfermedad de Parkinson**

Los síntomas de la Enfermedad de Parkinson pueden clasificarse en: motores y no motores, que según avanza la enfermedad estos empeoran progresivamente, en particular los síntomas motores [9].

#### **Clasificación de los síntomas de la EP**

Las características de cada uno de los síntomas se detallan a continuación [9]:

##### **Síntomas motores**

- Temblor en reposo, que se presentan en las primeras etapas de la enfermedad.
- Bradicinesia, significa el movimiento lento que describe una reducción general del movimiento espontáneo.
- Rigidez, inflexibilidad en las extremidades, cuello y tronco.
- Inestabilidad postural, una tendencia a ser inestable cuando se coloca en vertical.
- Falta de expresión facial.

##### **Temblor**

Se define al temblor como una oscilación involuntaria. La mayoría de los temblores son en los brazos. Sin embargo, pueden afectar a una o más partes del cuerpo, incluso la cabeza o la voz. Se puede dar en personas sanas, pero existen varios trastornos que



pueden contribuir a su desarrollo, en su mayoría de origen neurológico. Entre estas enfermedades, la de Parkinson es la más frecuente [18] .

Existen diferentes tipos de temblores los cuales se detallan en la tabla 1.

**Tabla 1. Clasificación del temblor [18]**

<b>TIPO</b>	<b>DEFINICIÓN</b>
<b>Temblor en reposo</b>	La persona está acostada o sentada y apoya la parte del cuerpo que está temblando. No es necesario ni se trata de activar los músculos esqueléticos. Normalmente, la frecuencia de estos temblores es de 3 a 6 Hz.
<b>Temblor postural</b>	El temblor se produce cuando la persona adopta una postura sostenida contra la fuerza de la gravedad sin realizar ningún tipo de movimiento, como por ejemplo cuando se extiende el brazo o se señala algo o a alguien. En esta parte del cuerpo es donde se produce el temblor. La frecuencia de este tipo de temblores suele ser de 4 a 12 Hz.
<b>Temblor cinético</b>	Este tipo de temblor se produce cuando la persona quiere realizar un movimiento consciente y voluntario, como hablar, comer, escribir, etc. Se produce una contracción voluntaria de los músculos esqueléticos. La frecuencia de este tipo de temblores suele ser de 2 a 7 Hz.

**Elaborado por: el investigador**

### **Síntomas no motores**

- Pérdida del sentido del olfato.
- Estreñimiento, trastorno del sueño REM.
- Trastorno del estado de ánimo.
- Hipotensión ortostática (presión arterial baja al ponerse de pie). La hipotensión ortostática puede hacer que se sienta mareado o aturdido, e incluso puede provocar un desmayo.
- Alteraciones de la visión.

### **1.3.4 Tratamiento**

Hoy en día, la enfermedad de Parkinson (EP) no tiene una cura definitiva, pero sí se puede manejar de manera muy efectiva. El tratamiento va dirigido tanto al control de los síntomas motores como al de los no motores. El método más frecuente para controlar la enfermedad de Parkinson consiste en utilizar medicamentos, al inicio de la enfermedad predominan los síntomas motores que responden a levodopa [2]. Las medicinas que aportan levodopa al organismo mejoran todos los síntomas motores cardinales, ya que tienen la virtud de suplir el déficit del neurotransmisor dopamina [7].

Con el avance de la enfermedad se presentan mayores complicaciones sintomáticas que no responden a la medicación dopaminérgica, por ende, la intervención quirúrgica basada en la implantación de electrodos en los ganglios del cerebro sigue siendo una herramienta importante para mejorar los síntomas motores, pero está reservada para una pequeña minoría de pacientes menores de 70 años [8].

### **1.3.5 Terapias físicas o fisioterapia**

Además del tratamiento médico y quirúrgico, otra alternativa de tratamiento para la EP son las terapias físicas o fisioterapias que ayudan a minimizar los síntomas motores y no motores de los pacientes [8].

La fisioterapia se centra en seis áreas específicas esenciales: las transferencias, la postura, el funcionamiento de las extremidades superiores, el equilibrio, la marcha y la capacidad física, y la inactividad [8].

### **1.3.6 Terapia ocupacional**

La terapia ocupacional está enfocada en potenciar las habilidades que son más afectadas durante la enfermedad y mejorar o mantener la autonomía en el autocuidado (comida, vestido y aseo), las actividades productivas y de ocio [8].

Trabaja la coordinación motora fina y gruesa, la fuerza, la movilidad articular, la atención, la memoria, la concentración y la resolución de problemas, para lo cual se

colabora e interviene de forma conjunta con otras disciplinas, especialmente con psicología (terapia cognitiva) [18].



**Figura 1. Juego de terapia ocupacional**

**Fuente [18]**

### **Control de la motricidad fina y gruesa**

Se refiere a la habilidad de coordinar los movimientos musculares precisos que tienen lugar en áreas del cuerpo como los dedos. Un ejemplo de control de la motricidad fina sería recoger un objeto pequeño con el dedo índice y el pulgar. La rigidez y el temblor obstaculizan esta habilidad manual, impidiendo que acciones que se necesiten precisión con los dedos sean más complicada o imposibles para la persona con la EP [19]. El control de la motricidad gruesa los grandes movimientos que se efectúan con piernas, brazos, pies o cuerpo entero.

### **1.3.7 Logopedia**

El objetivo de la terapia logopedia es mejorar la comprensibilidad del habla, la comunicación y el lenguaje a través de técnicas de deglución o herramientas que incluyen ejercicios respiratorios, tratamientos de la voz, y otros, como son los aparatos de comunicación, como tableros de letras o de estimulación, amplificadores de voz, sistemas de voz digitalizada, mensajes de voz grabados, retroalimentación auditiva retrasada o dispositivos de retroalimentación con un ordenador portátil [8].

### **1.3.8 Sensores**

Un sensor es un aparato que puede detectar diferentes magnitudes, tanto físicas como químicas, y convertirlas en señales eléctricas. Las magnitudes que pueden ser medidas incluyen la temperatura, la intensidad de la luz, la distancia, la aceleración, la inclinación, el desplazamiento, la presión, la fuerza, la torsión, la humedad, el movimiento, el pH, entre otros. La señal eléctrica resultante puede ser una resistencia eléctrica, una capacitancia eléctrica, una tensión eléctrica, una corriente eléctrica, etc. [20]

#### **Acelerómetro**

Este tipo de sensor transforma la aceleración en una señal eléctrica proporcional a la magnitud de dicha aceleración. En la actualidad existen los acelerómetros triaxiales que son capaces de registrar movimientos en diferentes planos del espacio, siendo más sensibles y capaces de registrar el tiempo que se emplean diferentes actividades. El módulo Acelerómetro MPU tiene un giroscopio de tres ejes con el que podemos medir velocidad angular y un acelerómetro también de 3 ejes con el que medimos los componentes X, Y y Z de la aceleración [21].

La dirección de los ejes está representada en el módulo, la cual hay que tener en cuenta para no equivocarnos en el signo de las aceleraciones [22].

#### **Giroscopio**

El sensor giroscópico es un dispositivo que puede medir y mantener la orientación y la velocidad angular de un objeto. Son más avanzados que los acelerómetros. Estos pueden medir la inclinación y la orientación lateral del objeto mientras que el acelerómetro sólo puede medir el movimiento lineal [23].

Los sensores giroscópicos son útiles para determinar la rotación de un objeto. A diferencia de los humanos, estos sensores son capaces de medir la velocidad angular, que se refiere al cambio en el ángulo de giro de un objeto por unidad de tiempo en grados por segundo. Se utilizan en aplicaciones en las que la orientación del objeto es difícil de percibir por los humanos. [23].

### **1.3.9 Almacenamiento en la nube**

Este servicio ofrece accesibilidad desde cualquier lugar y en cualquier momento, ya que los datos están almacenados en servidores remotos administrados por un proveedor de almacenamiento en la nube. Además, el almacenamiento en la nube suele ser más seguro y asequible que el almacenamiento local, ya que los proveedores suelen tener medidas de seguridad y copias de seguridad avanzadas. Por esta razón, es una solución popular para el almacenamiento de datos y documentos importantes. Existen diferentes tipos de almacenamiento en la nube, incluyendo almacenamiento personal para usuarios individuales para guardar copias de seguridad de correo electrónico, fotos, videos, y otros archivos personales, así como también almacenamiento empresarial para las compañías, que les permite transferir y almacenar de manera segura sus archivos de datos y compartirlos entre diferentes ubicaciones. [24].

## **1.4 Objetivos**

### **1.4.1 Objetivo general**

Implementar un sistema electrónico de rehabilitación y monitoreo para personas que padecen la Enfermedad de Parkinson.

### **1.4.2 Objetivos específicos**

- Determinar las características de la Enfermedad de Parkinson (EP) y los síntomas que presenta en las extremidades superiores.
- Identificar las diferentes técnicas de rehabilitación en Pacientes con la EP y los recursos utilizados.
- Diseñar un sistema electrónico para estimular la rehabilitación de las extremidades superiores de pacientes que sufren la EP.

## **CAPÍTULO II**

### **METODOLOGÍA**

#### **2.1 Materiales**

En la implementación del presente proyecto, se emplearon varios componentes electrónicos como: switch para el encendido y apagado del sistema, pulsadores para seleccionar al Paciente a monitorear de acuerdo con el identificador único, led verde para indicar cuando el dispositivo se conecta correctamente a una red Wi-Fi, un módulo TP4056 que está encargado de la carga de la batería, posterior a esto se utilizó una tarjeta ESP8266 como microcontrolador principal y sensor MPU6050 para medir la aceleración del movimiento en los tres ejes X, Y y Z. Para lo que viene siendo el sistema de rehabilitación se utilizó un microcontrolador NodeMCU, interruptores de posición para controlar la ubicación de las esferas metálicas que son transportadas por el paciente y una LCD 16x2 el cual muestra el tiempo empleado en cada prueba.

Para el desarrollo del Dashboard destinado a mostrar todos los datos obtenidos durante el monitoreo y rehabilitación de los pacientes, se empleó el software Node-Red el cual recopila los datos enviados hacia el Bróker de Mosquitto desde el microcontrolador ESP8266, estos valores son almacenados en MariaDB un sistema gestor de base de datos, además, los datos son presentados en diferentes interfaces gráficas como histogramas, gráficos de tipo guage y tablas de datos. Todos estos softwares están instalados en un servidor de Ubuntu el cual está alojado en Microsoft Azure.

#### **2.2 Métodos**

##### **2.2.1 Modalidad de Investigación**

###### **Investigación Aplicada**

El presente proyecto se sustentó en una Investigación Aplicada, porque se emplearon los conocimientos adquiridos durante la carrera y se realizaron investigaciones que apoyaron la realización del proyecto; estas investigaciones se lo obtuvieron tanto de las bases de datos como de las bibliotecas virtuales, los mismos que son proporcionados por la Universidad.

## **Investigación Bibliográfica**

Investigación Bibliográfica, porque el proyecto de investigación se sustentó mediante la recopilación de información de tesis, artículos científicos, publicaciones en revistas y libros, relacionados a temas acerca de pacientes con Enfermedad de Parkinson y sus procesos de rehabilitación.

## **Investigación de Campo**

Investigación de Campo, porque el proyecto desarrollado implicó la participación del paciente y, mediante el sistema electrónico se podrá monitorear y realizar las terapias de extremidades superiores del paciente con EP.

### **2.2.2 Recolección de Información**

Para lograr la recolección de información se emplearon libros, revistas, fuentes online y proyectos desarrollados, asimismo datos de incidencia y prevalencia de la EP, por lo que se tomará en cuenta fuentes de información confiables que permitan el desarrollo del proyecto.

### **2.2.3 Procesamiento y Análisis de Datos**

Para el procesamiento y análisis de datos se realizó los siguientes pasos:

- Análisis de todas las fuentes de información documental que tengan relación con el diseño de un sistema de rehabilitación para personas con la Enfermedad de Parkinson.
- Estudio de las propuestas de solución planteadas para el monitoreo y rehabilitación de pacientes con Enfermedad de Parkinson.
- Planteamiento de la propuesta de solución.
- Control y verificación de los datos obtenidos mediante el testeado del dispositivo.

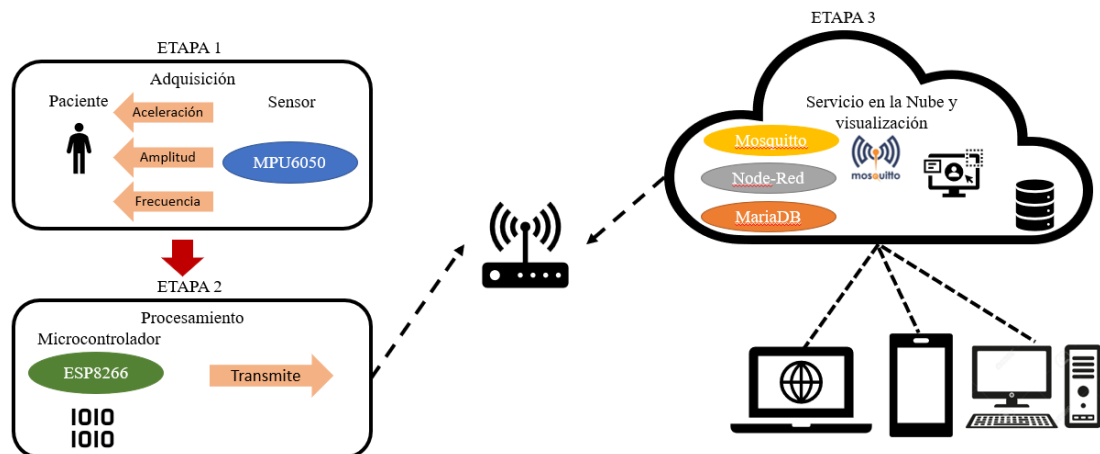
## CAPITULO III RESULTADOS Y DISCUSIÓN

### 3.1 Análisis y discusión de los resultados

#### 3.1.1 Desarrollo de la propuesta

Para dar cumplimiento con los objetivos planteados del sistema electrónico de rehabilitación y monitoreo para personas que padecen la enfermedad de Parkinson se implementó un prototipo que adquiere señales de un paciente, tales como la amplitud, aceleración y la frecuencia, el monitoreo se realiza en tiempo real.

Se construyó un dispositivo electrónico portable con el fin de obtener todos los datos mencionados anteriormente. La arquitectura de este dispositivo se puede observar en la figura 2, el mismo que consta de 3 etapas: de adquisición, de procesamiento y de servicio en la nube y visualización.



**Figura 2. Arquitectura del dispositivo de monitoreo de las señales del Parkinson**  
Elaborado por: el investigador

#### 3.1.2 ETAPA 1.- Adquisición


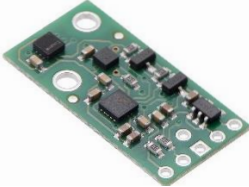

Para la elaboración del dispositivo se ha seleccionado cada elemento electrónico de acuerdo con los requerimientos necesarios para la obtención de los resultados esperados.



### 3.1.2.1 Selección de sensores

En la tabla 2 se representa la comparación realizada de los sensores disponibles en el mercado y con las características más importantes acorde al propósito del proyecto. Se escogió el sensor MPU6050 el cual cuenta con un acelerómetro y giroscopio, además de contar con un mayor número de ejes en comparación con el sensor ADXL345 ya que este no cuenta con un giroscopio. Por otro lado, el precio del prototipo es muy económico con respecto a los otros sensores. El uso de este sensor permitirá determinar la aceleración en el movimiento en los 3 ejes, esta aceleración será procesada para poder determinar la amplitud y frecuencia del movimiento de la mano.

Tabla 2. Comparación de los sensores inerciales [25]

Características	Sensor <b>ADXL345</b>	Sensor <b>AltIMU-10</b>	Sensor <b>MPU6050</b>
<b>Figura</b>			
<b>Dimensiones</b>	2.1 cm largo, 1.58 cm ancho y 0.3 cm espesor	2.1 cm largo, 1.3 cm ancho y 0.3 cm espesor	2 cm largo, 1.6 cm ancho y 0.3 cm espesor
<b>Voltaje de operación</b>	2 – 3.6 V	2.5 – 5.5 V	3.3 – 5 V
<b>Rango acelerómetro</b>	2g/4g/8g/16g	2g/4g/8g/16g	2g/4g/8g/16g
<b>Rango giroscopio</b>	No incluye	(250/500/2000) grad/seg	(250/500/1000/2000) grad/seg
<b>Ejes</b>	3	6	6
<b>Interfaz</b>	SPI e I2C	I2C	I2C
<b>Costo</b>	\$12.00	\$22.00	\$7.00

Elaborado por: investigador

### 3.1.2.2 Alimentación del dispositivo

Tomando en cuenta que se planteó desarrollar un dispositivo inalámbrico y portátil, se identificó y se seleccionó dos elementos necesarios que puedan alimentar y mantener funcionando al dispositivo sin la necesidad de estar conectado de forma permanente a una fuente de alimentación. Los dispositivos seleccionados fueron una batería y un módulo de carga.

#### Batería

Para la alimentación del dispositivo se ha optado por una batería recargable de iones de litio, la misma que se muestra en la figura 3, la batería alcanza un voltaje de 3.7V a 4.2V y una corriente de 1500mAh y está constituida por varias celdas lo que permitió almacenar una gran cantidad de energía. Sus características se detallan en la tabla 3.



Figura 3. Batería Li – Ion

Fuente: [26]

Tabla 3. Especificaciones técnicas de la batería [26]

Características técnicas de la batería	
Modelo	18650 1500mah 3,7V
Tensión nominal (v)	3,7V
Capacidad nominal	1500mah
Tiempos de carga	500 veces
Vida útil	1 año

Elaborado por: Investigador

## Módulo de carga TP4056

Para cargar la batería de ion de Litio de 3.7V, se ha optado por el módulo de carga TP4056, ya que se ajusta a las necesidades del dispositivo por sus dimensiones, voltaje de entrada, interfaz de entrada, corriente y tipo de batería a cargar. El módulo de carga se puede observar en la figura 4 y sus características de detallan en la tabla 4.

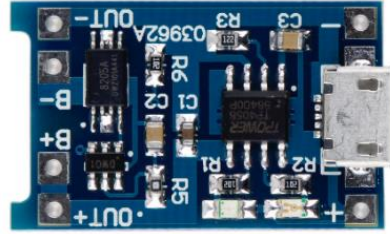


Figura 4. Modulo 4056 cargador de batería

Fuente: [27]

Tabla 4. Especificaciones técnicas modulo TP4056 [27]




Características del Módulo TP4056	
Voltaje de entrada	4,5 a 5,5 V
Interfaz de entrada	Mini USB
Temperatura de trabajo	-10 °C a 85 °C
Dimensión	28 mm * 19 mm * 6 mm
Cargador de	LiPo o Li-ion de una sola celda de 3.7V 1 Ah
Corriente de carga	1 <sup>a</sup>

### 3.1.3 ETAPA 2.- Procesamiento y transmisión

#### 3.1.3.1 Selección del microcontrolador

El procesamiento y adquisición de datos es fundamental para el funcionamiento del prototipo debido a que permite tomar las señales analógicas y transformarlas a señales digitales para poder ser procesadas mediante un microcontrolador, el cual permite gestionar dichas señales para su interpretación y visualización. En la tabla 5 se presenta las características de los tres microcontroladores más acordes con los requerimientos establecidos para dar cumplimiento con los objetivos del proyecto.

Tabla 5. Comparación de los microcontroladores [28]

Parámetros	ATMEGA2560	PIC16F887	ESP8266
Figura			
Frecuencia de operación	16 MHz	0 – 20 MHz	80 MHz
Número de entradas digitales	54	21	9 pines GPIO
Voltaje de entrada	7 – 12 V	5.5 V	5 V
Temperatura de operación	-45 a 85 °C	-45 a 150 °C	Máxima de 125 °C
Memoria interna	256.8 KB RAM	8 KB, 256 bytes de EEPROM	4MB de memoria FLASH (32 Mbit)
Número de entradas analógicas	16	14	1
Tipo de comunicación	SERIE UART-12C2SPI, RS232	RS-485 RS-232 Y LIN2.0	I2C, SPI, WIFI 802.11 b/g/n
Consumo de corriente	250 mAh	220 uAh	80 mAh

Elaborado por: investigador

La principal desventaja que presentan los microcontroladores ATMEGA2560 y PIC16F887 en comparación con la ESP8266 es que no cuentan con la comunicación Wi-Fi que es imprescindible para el desarrollo del prototipo. Además, su consumo de corriente es de 80 mA respecto a los otros dispositivos. Por otra parte, la tarjeta ESP8266 cuenta con un tamaño compacto y un número de pines exacto para la realización del proyecto, soporta una mayor temperatura de operación, gran variedad de tipo de comunicación, una memoria interna superior. Por esta razón, se optó por utilizar la tarjeta ESP8266 como microcontrolador principal en este proyecto.



**Figura 5. Tarjeta ESP8266**

**Fuente: [28]**

### ***3.1.3.2 Selección de la tecnología inalámbrica***

En la actualidad existe una gran cantidad de tecnologías inalámbricas idóneas para realizar este proyecto; el uso de la tecnología Bluetooth se descarta debido a que su alcance es de 10 m. El uso de la tecnología ZigBee supone costos más elevados por lo que de igual manera se descarta. Wi-Fi posee un alcance más amplio respecto a conexión, es una normal de la IEEE llamada 802.11; además, existe una gran variedad de dispositivos que utilizan este estándar. Por otro lado, la tecnología GSM/GPRS tiene un rango de cobertura superior a las tecnologías antes mencionadas; sin embargo, para utilizar este servicio se debe contar con una tarjeta SIM con un plan celular de datos para la comunicación, lo que conlleva al usuario a un costo elevado en su uso.

**Tabla 6. Comparación de las tecnologías inalámbricas [29]**

<b>Parámetro</b>	<b>Bluetooth</b>	<b>ZigBee</b>	<b>Wi-Fi</b>	<b>GSM-GPRS</b>
<b>Estándar</b>	802.15.1	802.15.4	802.11	GSM
<b>Frecuencia de la señal</b>	2.4 GHz	2.4 GHz 915 MHz 868 MHz	2.4 – 5 GHz	300 – 1200 MHz
<b>Ancho de banda</b>	1 Mbps	20 – 250 Kbps	2 – 100 Mbps	20 – 100 Mbps
<b>Número de nodos por red</b>	7	65000	255	Dependiendo de la operadora celular
<b>Distancia de transmisión sin interferencias</b>	10 – 100 m	1 – 75 m	50 - 100 m	Depende de la operadora
<b>Memoria</b>	100 KB	32 - 60 KB	100 KB	Depende de los dispositivos que se utilicen
<b>Topología</b>	Punto a punto, estrella y árbol	Punto a punto, estrella, malla, árbol y cluster	Bus, anillo, estrella y árbol	Topología celular

**Elaborado por: investigador**

Por lo tanto, para realizar el proyecto se optó por utilizar la tecnología Wi-Fi ya que es muy utilizado para conectarse a internet de forma sencilla y por su amplia gama de equipos y su utilización en aplicaciones IoT.

### **3.1.4 ETAPA 3.- Servicio en la nube y visualización**

#### **3.1.4.1 Selección del servidor**

Para seleccionar e instalar el servidor que está destinado a recopilar los datos enviados del controlador se realizó un estudio previo de las plataformas más comunes de

creación de servidores web en la nube. En la tabla 7, se muestran las características principales de cada una de las plataformas.

**Tabla 7. Comparación de las plataformas Cloud Computing [30]**

<b>Parámetros</b>	<b>Microsoft Azure</b>	<b>Google Cloud Platform</b>	<b>Amazon EC2 - AWS</b>
<b>Soporte disponible</b>	Foros, comunicaciones telefónicas, chats en vivo, documentación.	Foros, documentación, preparación, comunicaciones telefónicas	Foros, documentación, herramientas de aprendizaje.
<b>Bases de datos</b>	Azure SQL	Cloud SQL Cloud Spanner Cloud Bigtable Cloud Firestore Cloud Momorystore Firebase Realtime Database	MySql PostgreSQL MariaDB MongoDB Redis Memcached
<b>Sistema Operativo del servidor</b>	Linux, Windows	Linux, Windows	Linux, Windows
<b>Preconfiguración del sistema operativo</b>	Azure SQL	Ubunto Centos OS FreeBSD Oracle Linux Windows server OpenSUSE Linux	MongoDB MySql PostgreSQL Redis Memcached MariaDB

Elaborado por: investigador

Los servicios de Amazon web han estado operando desde 2006, siendo la plataforma más antigua con relación a Microsoft Azure que ha estado operando desde el 2010 y cuenta con una nube más extendida, mientras que la plataforma de Google Cloud se inicia en 2011 y desde sus inicios ha acumulado importantes servicios como Airbus, Coca-Cola, HTC, Spotify, etc. En cuanto a la información del manejo y el uso correcto de las plataformas (Google Cloud, Microsoft Azure, Amazon EC2) cada una cuenta con una gran cantidad de foros y documentación. En todo lo que corresponde las especificaciones hay que recalcar que AWS cuenta con más sistemas operativos preconfigurados, pero Microsoft Azure tiene el alcance más alto. Microsoft Azure ofrece más compatibilidad orientado al marco de autoaprendizaje. El precio de las tres plataformas va en función del uso que se le dé. Sin embargo, Microsoft Azure ofrece una ventaja la cual mediante el registro del correo institucional se puede contar con un crédito de \$100 durante un año, junto con el uso de los servicios gratuitos como parte del incentivo al aprendizaje, investigación sin fines comerciales o trabajos de diseño, desarrollo, pruebas y demostración de aplicaciones, lo cual es importante para la realización de este proyecto. En consecuencia, en el presente proyecto, se usó la plataforma en la nube Microsoft Azure, porque va acorde a las necesidades de este proyecto en cuanto a flexibilidad, soporte y seguridad.

#### ***3.1.4.2 Selección del protocolo de comunicación***

Se llevó a cabo un estudio de los protocolos más utilizados en Cloud Computing e Internet de las cosas para seleccionar el protocolo de comunicación adecuado. La tabla 8 presenta sus características más importantes. MQTT es un protocolo de comunicación que permite que varios clientes se comuniquen entre sí a través de un intercambio central y es eficiente como un bus de comunicaciones para datos en tiempo real. Por otro lado, CoAP transfiere información de estado de manera uno a uno entre el cliente y el servidor. HTTP tiene un costo más elevado en cuanto a el establecimiento de conexión en comparación con MQTT, pero no es común en aplicaciones IoT debido a que se envían y reciben múltiples mensajes en la misma sesión TCP, aprovechando las ventajas de MQTT.



**Tabla 8. Comparación de los protocolos de comunicación [31]**

<b>Parámetros</b>	<b>MQTT</b>	<b>CoAP</b>	<b>HTTP</b>
<b>Arquitectura</b>	Cliente/Broker	Cliente/Broker o Cliente/Servidor	Cliente/servidor
<b>Calidad de servicio (QoS)/Confiabilidad</b>	QoS 0: como máximo una vez (dispara y olvida) QoS 1: al menos una vez QoS 2: exactamente una vez	Mensaje confirmable (similar a como máximo una vez) o Mensaje no confirmable (similar a por lo menos una vez)	Limitado (a través del protocolo de transporte – TCP)
<b>Conectividad</b>	Uno-a-uno, uno-a-muchos y muchos-a-muchos	Uno a uno y muchos a muchos	Uno-a-uno
<b>Consumo de Energía</b>	MQTT es energéticamente más eficiente	CoAP es más eficiente en términos de energía	Requiere la mayor potencia/energía consumida por HTTP fue mucho mayor que con MQTT
<b>Consumo de banda ancha</b>	Consume más banda ancha	Implica el ancho de banda más bajo	Implica mayor ancho de banda
<b>Puerto predeterminado</b>	1883/ 8883 (TLS/SSL)	5683 (UDP Port)/ 5684 (DLTS)	80/ 443 (TLS/SSL)
<b>Aplicaciones</b>	Automatización del hogar, aplicaciones de nivel empresarial	Casas inteligentes, red inteligente y automatización de edificios	Web

Elaborado por: investigador

MQTT requiere que todos los clientes estén al tanto del formato de los mensajes para que se pueda llevar a cabo la comunicación, ya que no ofrece etiquetado de mensajes. Por otro lado, CoAP se destaca en este aspecto, ya que proporciona soporte para la negociación y el descubrimiento de contenidos, lo que permite que los dispositivos sepan cómo intercambiar información.

En este sentido, el protocolo de comunicación más adecuado para este proyecto es MQTT, ya que es más flexible y cumple con las necesidades de transmisión de datos en tiempo real de alto volumen.

### 3.1.4.3 Selección de la base de datos

**Tabla 9. Comparación de las bases de datos [32] [33]**

Parámetros	Bases de datos		
	MySQL	Oracle	PostgreSQL
<b>Descripción</b>	Sistema de manejo de bases de datos con código abierto que utiliza SQL.	Sistema de gestión de bases de datos relacionales	Sistema de gestión de base de datos objeto – relacional Basado en PostgreSQL, de código abierto.
<b>Arquitectura de Implementación</b>	Cliente / servidor	Cliente / Servidor	Cliente / Servidor
<b>Lenguaje de Programación</b>	C, C++	Visual C++	C/C++
<b>S.O. Compatible</b>	FreeBSD, Linux OS X, Solaris y Windows	Linux, Windows, Mac OS X y Unix	Linux, Windows y Mac OS.
<b>Compatibilidad con Node-RED</b>	Si	Si	Si
<b>Licencia</b>	Privada	Libre a nivel de usuario	Libre

Elaborado por: investigador

Se ha decidido tener un sistema que permita clasificar y organizar los datos según el identificador del paciente, es decir, los datos serán ordenados utilizando un identificador único para cada paciente. PostgreSQL es el sistema de gestión de bases de datos más robusto en el mercado, pero no es tan comúnmente utilizado como MySQL y cuenta con menos tutoriales y cursos disponibles en comparación con MySQL.

Por otro lado, Oracle es una de las bases de datos más utilizadas por las empresas, pero su costo es elevado, incluyendo el costo de la información. MySQL es un sistema de base de datos muy popular y ampliamente utilizado en todo el mundo, por lo que existe una gran cantidad de información y tutoriales disponibles, y es adecuado para proyectos pequeños y medianos.

Se ha elegido MySQL para este proyecto por ser fácil de usar para adquirir datos de dispositivos electrónicos, y puede ser conectado directamente con Node-RED para representar de manera dinámica los parámetros del sistema de monitoreo.

#### ***3.1.4.4 Selección de la herramienta de visualización***

Se ha elegido Node-RED como herramienta de visualización de los datos recopilados por el sistema de monitoreo por una serie de motivos. En primer lugar, Node-RED es altamente compatible con la base de datos MySQL seleccionada para almacenar la información del paciente. Esta compatibilidad permite una fácil integración entre los dos sistemas y una gestión eficiente de los datos.

Además, Node-RED es una plataforma de programación visual completamente gratuita, lo que significa que no hay costos adicionales asociados con su uso y está disponible para todas las personas interesadas en utilizarla.

La herramienta también es muy intuitiva y fácil de utilizar, lo que la hace ideal para visualizar de manera gráfica el flujo de información que proviene de los dispositivos electrónicos. Los datos se pueden representar en tiempo real y de manera dinámica, lo que permite una fácil interpretación de los resultados. En resumen, Node-RED es una excelente herramienta para visualizar los datos recopilados en el sistema de monitoreo y es por eso que se ha elegido para este proyecto.

### 3.1.5 Estancia en la Nube

Para el desarrollo de este proyecto, se utilizó el servicio de Azure Virtual Machines, que ofrece comodidad en la virtualización sin necesidad de tener un hardware físico en ejecución. Se empleó la suscripción de Azure Education, obtenida con un correo electrónico de una institución educativa, para poder utilizar los servicios en la nube de Microsoft Azure. Se desplegó una máquina virtual de Ubuntu Server, que aloja los servicios de Node-RED, Mosquitto y MariaDB.

### Servicios en la nube de Microsoft Azure

Para emplear los servicios de Microsoft Azure, se utilizó una cuenta registrada con un correo electrónico institucional, lo que permitió acceder a la suscripción de Azure Education. Esta suscripción cuenta con un crédito de \$100 por un año, lo que resultó suficiente para el desarrollo del proyecto.

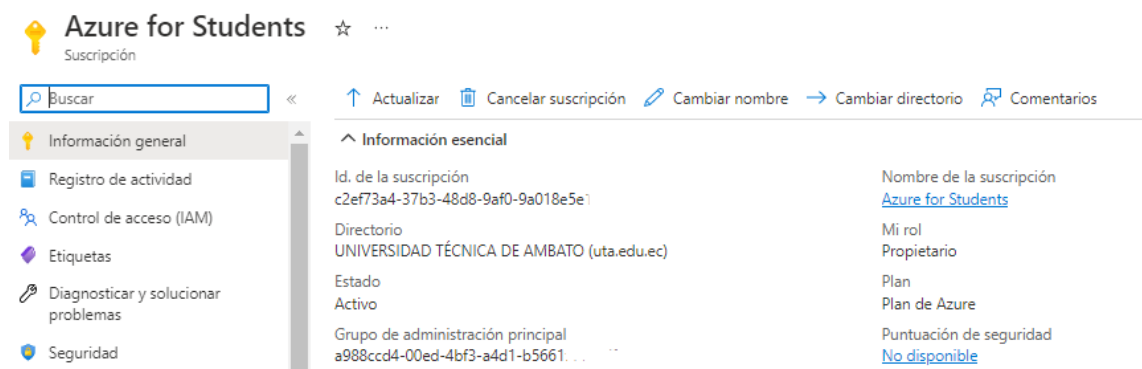


Figura 6. Suscripción de Azure Education

Elaborado por: investigador

Para poder utilizar los servicios de Azure fue necesario crear un grupo de recursos, como se puede observar en la figura 7.

### Crear un grupo de recursos

Datos básicos   Etiquetas   Revisar y crear

Grupo de recursos - Contenedor que incluye los recursos relacionados para una solución de Azure. El grupo de recursos puede contener todos los recursos de la solución o solamente los recursos que quiere administrar en grupo. Debe decidir cómo quiere asignar los recursos a los grupos de recursos según lo que resulte más pertinente para su organización. [Más información](#)

Detalles del proyecto

Suscripción \* ⓘ Azure for Students

Grupo de recursos \* ⓘ Proyecto

Detalles del recurso

Región \* ⓘ (US) Central US

**Figura 7. Creación del grupo de recursos**

**Elaborado por: investigador**

Una vez establecido un nombre para el grupo de recursos y elegida una región, se validaron los datos.

### Crear un grupo de recursos

✓ Validación superada.

Datos básicos   Etiquetas   Revisar y crear

Datos básicos

Suscripción	Azure for Students
Grupo de recursos	Proyecto
Región	Central US

Etiquetas

Ninguno

**Figura 8. Validación del grupo de recursos**

**Elaborado por: investigador**

Una vez finalizada la creación de grupo de recursos se puede utilizar los servicios de Azure.

## Servidor en la nube de Microsoft Azure

El servidor se encuentra instalado como una máquina virtual de Ubuntu Server en Microsoft Azure. En el servidor, se encuentra instalado Node-RED, una herramienta utilizada para comunicar hardware de forma rápida y sencilla. Node-RED se encarga de recopilar las métricas enviadas al Broker de Mosquito, que también está instalado en el mismo servidor, y que provienen del microcontrolador ESP8266 mediante conexión WiFi. Los datos se almacenan en una base de datos MariaDB. Luego, los datos son representados en gráficas en el nodo dashboard propio de Node-RED. El servidor funciona con el sistema operativo Ubuntu Server 20.04 LTS - Gen2, que es adecuado para el desarrollo del proyecto. Se instaló el servidor de Ubuntu Server de la siguiente forma:

**Crear una máquina virtual** ...

**⚠** Al cambiar opciones básicas se pueden restablecer las selecciones realizadas. Revise todas las opciones antes de crear la máquina virtual.

**Detalles del proyecto**  
Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción \* ⓘ Azure for Students

Grupo de recursos \* ⓘ Proyecto-LC  
[Crear nuevo](#)

**Detalles de instancia**

Nombre de máquina virtual \* ⓘ ServerParkinson

Región \* ⓘ (US) West US 3

Opciones de disponibilidad ⓘ Zona de disponibilidad

Zona de disponibilidad \* ⓘ Zonas 1  
**👤** Ahora puede seleccionar varias zonas. Si selecciona varias zonas, se creará una VM por zona. [Más información](#)

Tipo de seguridad ⓘ Estándar

[Revisar y crear](#) < Anterior Siguiendo: Discos >

**Figura 9. Creación del servidor Ubuntu Server**

**Elaborado por: investigador**

Para la implementación del servidor web, es necesario especificar el tipo de suscripción, el grupo de recursos, la región y zona en la que está disponible la creación del servidor, además de la memoria RAM y el número de núcleos del CPU. Se optó por utilizar 1 GB de memoria RAM y un núcleo para el CPU, ya que es suficiente para ejecutar los servicios necesarios para el desarrollo del proyecto.

⚠ Al cambiar opciones básicas se pueden restablecer las selecciones realizadas. Revise todas las opciones antes de crear la máquina virtual.

Imagen \*    
[Ver todas las imágenes](#) | [Configurar la generación de máquinas virtuales](#)

Arquitectura de VM  Arm64  x64

Ejecución de Azure Spot con descuento

Tamaño \*    
[Ver todos los tamaños](#)

Cuenta de administrador

Tipo de autenticación  Clave pública SSH  Contraseña

**i** Ahora, Azure genera automáticamente un par de claves SSH y le permite almacenarlo para usarlo en el futuro. Es una forma rápida, sencilla y segura de conectarse a la máquina virtual.

Nombre de usuario \*  ✓

Origen de clave pública SSH

**Revisar y crear** < Anterior Siguiente: Discos >

**Figura 10. Características del servidor web**

**Elaborado por: investigador**

Para acceder al servidor web se utiliza la autenticación de clave publica SSH.

Crear una máquina virtual ...

⚠ Al cambiar opciones básicas se pueden restablecer las selecciones realizadas. Revise todas las opciones antes de crear la máquina virtual.

Nombre de usuario \*  ✓

Origen de clave pública SSH

Nombre de par de claves \*  ✓

Reglas de puerto de entrada

Seleccione los puertos de red de máquina virtual que son accesibles desde la red Internet pública. Puede especificar acceso de red más limitado o granular en la pestaña Red.

Puertos de entrada públicos \*  Ninguno  Permitir los puertos seleccionados

Seleccionar puertos de entrada \*

**⚠ Esto permitirá que todas las direcciones IP accedan a la máquina virtual.**  
 Esto solo se recomienda para las pruebas. Use los controles avanzados de la pestaña Redes a fin de crear reglas para limitar el tráfico entrante a las direcciones IP conocidas.

**Revisar y crear** < Anterior Siguiente: Discos >

**Figura 11. Configuración de la autenticación SSH**

**Elaborado por: investigador**

Una vez implementada la máquina virtual, se pueden ver las características que tiene el servidor web y su dirección IP pública. Esta información es importante para conocer el rendimiento del servidor y tener acceso a él desde cualquier lugar con acceso a internet. Además, esto permite monitorear y realizar cambios en el servidor de manera remota, lo cual es una gran ventaja para el desarrollo y mantenimiento del proyecto.

The screenshot displays the configuration details for a virtual machine in the Azure portal. It is organized into two main sections: 'Información esencial' (Essential Information) and 'Propiedades' (Properties).

**Información esencial:**

- Grupo de recursos (mover): [proyecto-lc](#)
- Estado: Detenido (desasignado)
- Ubicación: West US 3 (Zona 1)
- Suscripción (mover): [Azure for Students](#)
- Id. de suscripción: c2ef73a4-37b3-48d8-9af0-9a018e5e19a7
- Zona de disponibilidad: 1
- Etiquetas (editar): [Haga clic aquí para agregar etiquetas.](#)
- Sistema operativo: Linux
- Tamaño: Standard B1s (1 vcpu, 1 GiB de memoria)
- Dirección IP pública: [20.38.46.67](#)
- Red virtual/subred: [Proyecto-LC-vnet/default](#)
- Nombre DNS: [Sin configurar](#)

**Propiedades:**

- Máquina virtual:**
  - Nombre del equipo: ServerParkinson
  - Estado de mantenimiento: -
  - Sistema operativo: Linux
  - Publicador: canonical
  - Oferta: 0001-com-ubuntu-server-focal
  - Plan: 20\_04-lts-gen2
  - Generación de VM: V2
- Redes:**
  - Dirección IP pública: [20.38.46.67](#)
  - Dirección IP pública (IPv6): -
  - Dirección IP privada: 10.0.0.4
  - Dirección IP privada (IPv6): -
  - Red virtual/subred: [Proyecto-LC-vnet/default](#)
  - Nombre DNS: [Configurar](#)

**Figura 12. Servidor web creado**

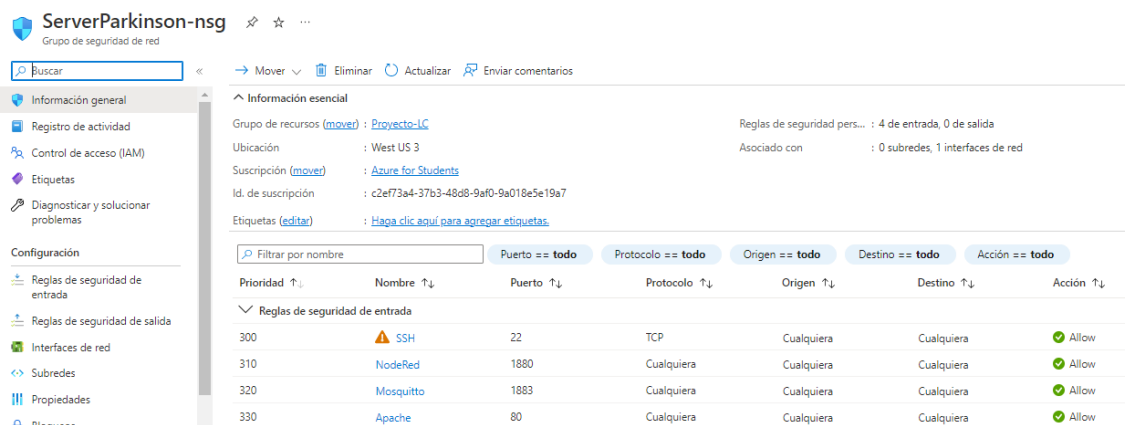
**Elaborado por: investigador**

Una vez que se descargó la llave de acceso a través de SSH, se pudo conectar al servidor virtual a través de una conexión segura utilizando PUTTY. Después de la conexión, se pudo acceder a la máquina virtual y realizar todas las configuraciones necesarias para el correcto funcionamiento de los servicios de Node-RED, Mosquitto y MariaDB. De esta manera, se garantizó la seguridad y la confiabilidad del proyecto, permitiendo un monitoreo eficiente de los datos provenientes de los dispositivos electrónicos.

## Habilitación de los puertos en Azure

Para utilizar los servicios de Node-Red, Mosquitto, MariaDB y Apache, se habilitaron los puertos correspondientes en la sección de reglas de seguridad del servidor web en Azure.





**Figura 13. Habilitación de puertos en Microsoft Azure**

**Elaborado por: investigador**

### 3.1.6 Instalación del Bróker Mosquitto

El Bróker de Mosquitto actúa como intermediario de mensajes de código abierto y pertenece a la Fundación Eclipse. Trabaja bajo el protocolo MQTT (Message Queuing Telemetry Transport), en el que dispositivos pueden publicar sus mensajes o suscribirse a un tópico. Es un software ligero y es perfecto para computadoras de bajo consumo y para servidores potentes. Para instalarlo, se utilizaron los siguientes comandos:

1. Primero se realizó una actualización de los paquetes de Ubuntu utilizando los siguientes comandos:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

2. Después se procedió con la instalación del bróker de Mosquitto desde los repositorios previamente actualizados, utilizando el siguiente comando:

```
sudo apt-get install mosquitto mosquitto-clients
```

3. Para comprobar que la instalación se realizó correctamente y que el servicio este ejecutándose se utilizó el comando:

```
service mosquitto status
```

```
● mosquitto.service - Mosquitto MQTT v3.1/v3.1.1 Broker
  Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor pre>
  Active: active (running) since Sun 2022-12-11 15:17:52 UTC; 58min ago
  Docs: man:mosquitto.conf(5)
        man:mosquitto(8)
  Main PID: 709 (mosquitto)
  Tasks: 3 (limit: 1076)
  Memory: 2.4M
  CGroup: /system.slice/mosquitto.service
          └─709 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

Dec 11 15:17:52 ServerParkinson systemd[1]: Starting Mosquitto MQTT v3.1/v3.1.1 >
Dec 11 15:17:52 ServerParkinson mosquitto[709]: [ 14.848442]~DLT~ 709~INFO >
Dec 11 15:17:52 ServerParkinson systemd[1]: Started Mosquitto MQTT v3.1/v3.1.1 >
lines 1-14/14 (END)
```

Figura 14. Bróker de Mosquitto funcionando correctamente

Elaborado por: investigador

### 3.1.7 Instalación de MariaDB

Utilizando el terminal de PUTTY el cual está conectado al servidor web de Ubuntu Server se instaló el servicio de MariaDB utilizando los siguientes comandos:

1. Se instaló MariaDB con el siguiente comando:

```
sudo apt-get install mariadb-server
```

2. Una vez instalado se habilitó el servicio y se inicia el servicio de MariaDB:

```
sudo systemctl enable mariadb
```

```
sudo systemctl restart mariadb
```

3. Para verificar que el servicio de MariaDB está funcionando correctamente, se utilizó el siguiente comando:

```
sudo systemctl status mariadb
```

```
● mariadb.service - MariaDB 10.3.37 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese
   Active: active (running) since Sun 2022-12-11 15:17:55 UTC; 56min ago
     Docs: man:mysql(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 708 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/>
   Process: 739 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_STA>
   Process: 771 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && v>
   Process: 1017 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_S>
   Process: 1019 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/>
   Main PID: 871 (mysqld)
     Status: "Taking your SQL requests now..."
       Tasks: 31 (limit: 1076)
      Memory: 113.5M
   CGroup: /system.slice/mariadb.service
           └─871 /usr/sbin/mysqld

Dec 11 15:17:52 ServerParkinson systemd[1]: Starting MariaDB 10.3.37 database s>
Dec 11 15:17:54 ServerParkinson mysqld[871]: 2022-12-11 15:17:54 0 [Note] /usr/>
Dec 11 15:17:55 ServerParkinson systemd[1]: Started MariaDB 10.3.37 database se>
Dec 11 15:17:55 ServerParkinson /etc/mysql/debian-start[1024]: Upgrading MySQL >
Dec 11 15:17:55 ServerParkinson /etc/mysql/debian-start[1054]: Checking for ins>
Dec 11 15:17:55 ServerParkinson /etc/mysql/debian-start[1058]: Triggering myis>
```

Figura 15. Servicio de MariaDB funcionando correctamente

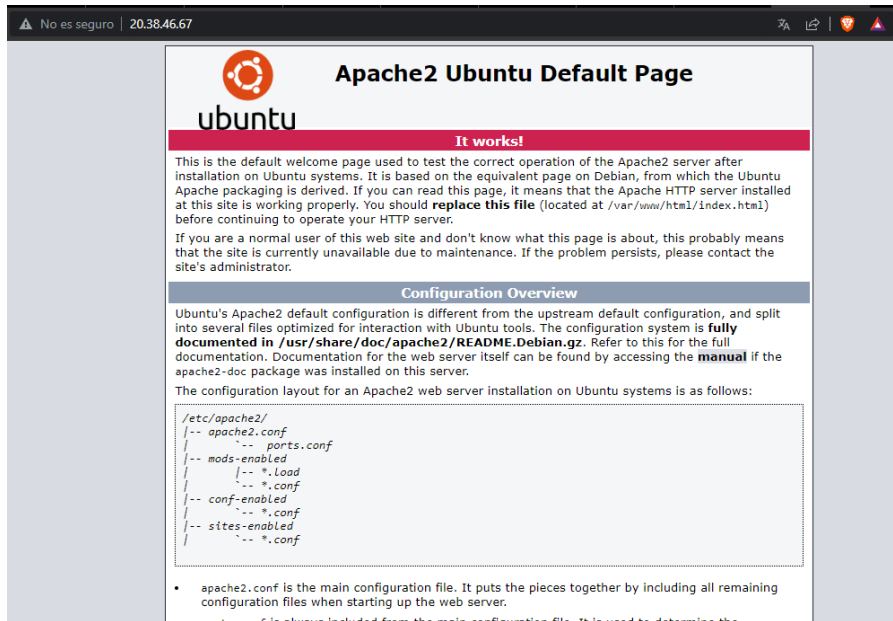
Elaborado por: investigador

4. Finalmente, se agregó una contraseña y se realizó una pequeña configuración con el siguiente comando:  
sudo mysql\_secure\_installation

### 3.1.8 Instalación del servidor Apache

Apache es un servidor web muy popular que se utiliza para alojar todos los archivos necesarios para visualizar un sitio web. La instalación de Apache es esencial para la gestión de la base de datos MariaDB y se realizó de la siguiente manera:

1. Primero, se instaló el software Apache con el siguiente comando:  
sudo apt-get install apache2
2. Luego, se habilitó y reinició el servicio de Apache con los siguientes comandos:  
sudo systemctl enable apache2  
sudo systemctl restart apache2
3. Finalmente, para verificar que el servicio de Apache está funcionando correctamente, se ingresó la dirección IP (20.38.46.67) del servidor en cualquier navegador. Si el servidor está configurado correctamente, se mostrará un mensaje en la página como se muestra en la figura 16.



**Figura 16. Servidor Apache funcionando correctamente**

**Elaborado por: investigador**

### **3.1.9 Instalación de PHP MyAdmin para administración de la base de datos**

Este software permite la administración de MariaDB utilizando una interfaz gráfica de fácil uso para los usuarios. La instalación se realizó con el siguiente comando:

```
sudo apt install phpMyAdmin
```

Para acceder a phpMyAdmin en Ubuntu Server, se necesitó realizar una configuración adicional en el servidor Apache. Se editó el archivo de configuración con el siguiente comando:

```
sudo nano /etc/apache2/apache2.conf
```

Se abrió un archivo en el cual se agregó la siguiente línea de código:

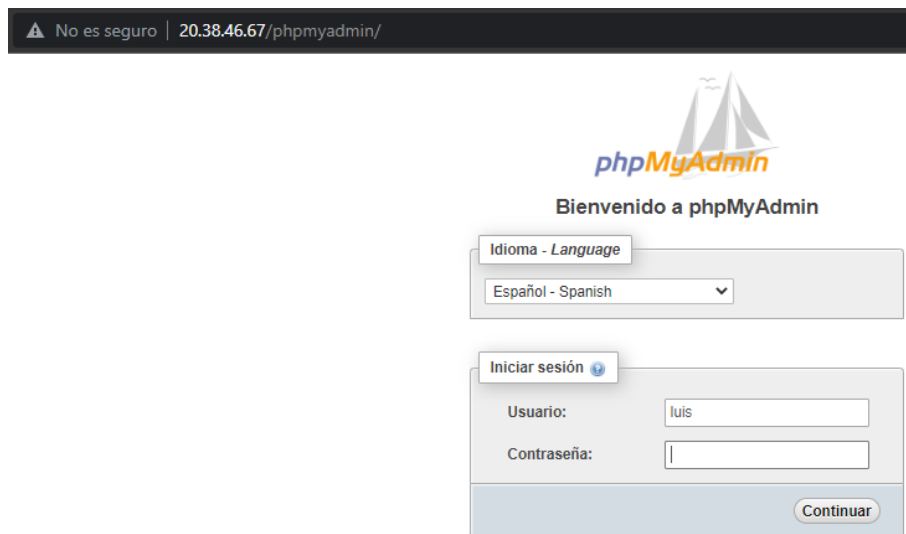
```
Include /etc/phpmyadmin/apache.conf
```

Se guardaron las configuraciones con 'Ctrl + x' y la tecla 'y'. Al terminal de guardar el archivo se reinició el servicio de Apache con el siguiente comando:

```
sudo service apache2 reload
```

Para verificar que el servicio de phpMyAdmin está funcionando correctamente, se accedió en el navegador a la dirección:

`http://20.38.46.67/phpmyadmin.`



**Figura 17. phpMyAdmin en el Navegador**

**Elaborado por: investigador**

### **3.1.10 Instalación de Node-Red para creación y acceso a la interfaz gráfica**

Se utilizó la terminal de PUTTY para poder instalar Node-Red en Ubuntu Server, mediante los siguientes comandos:

1. Primero, se instalaron node.js y Node-RED con los siguientes comandos:  

```
curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -  
  
sudo apt-get install -y nodejs build-essential  
  
sudo npm install -g --unsafe-perm node-red
```
2. Para que Node-Red se inicie automáticamente cada vez que se inicie el servidor, se utilizó PM2:  

```
sudo npm install -g --unsafe-perm pm2  
  
pm2 start `which node-red` -- -v  
  
pm2 save  
  
pm2 startup
```

```
[PM2] Spawning PM2 daemon with pm2_home=/root/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /usr/bin/node-red in fork_mode (1 instance)
[PM2] Done.
```

id	name	mode	Ⓢ	status	cpu	memory
0	node-red	fork	0	online	0%	25.9mb

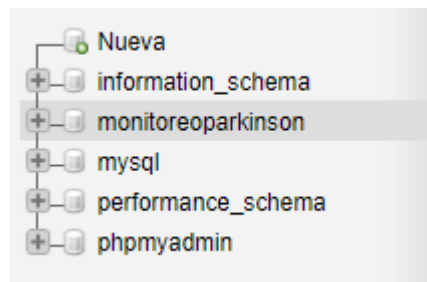
```
root@ServerParkinson:~# pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /root/.pm2/dump.pm2
```

**Figura 18. pm2 instalado correctamente y ejecutando Node-Red**

Elaborado por: investigador

### 3.1.11 Conexión con MySql

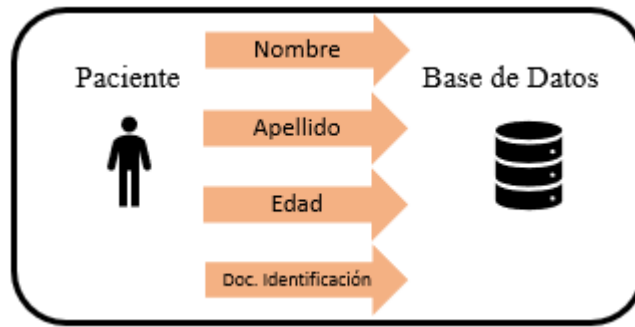
Para almacenar los datos provenientes del sensor, se necesita una base de datos que permita llevar a cabo un registro de todos los valores por fechas. Para ello, se utilizó el gestor de base de datos phpMyAdmin. El primer paso que se realizó fue crear una base de datos con el nombre de "monitoreoparkinson". Luego, dentro de esta base de datos, se crearon las tablas necesarias para almacenar los valores.



**Figura 19. Base de datos**

Elaborado por: investigador

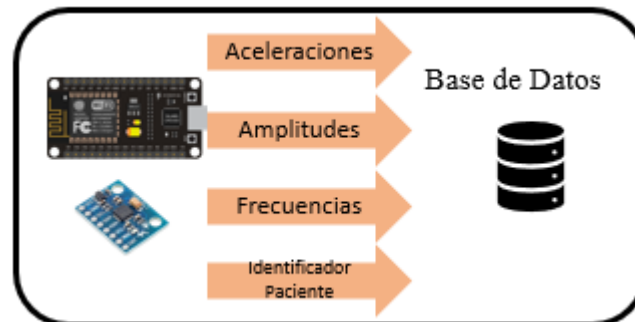
La primera tabla utilizada para almacenar los datos de los pacientes ingresados se llama "pacientes". En esta tabla se registran los siguientes datos de los pacientes: nombre, apellido, edad, documento de identificación y fecha en la que se llevó a cabo el registro.



**Figura 20. Almacenamiento de los datos del paciente**

**Elaborado por: investigador**

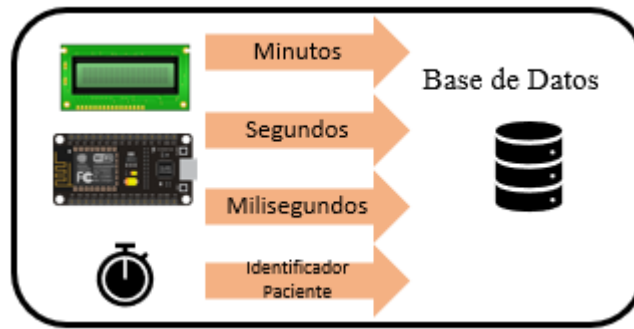
La segunda tabla se utilizó para almacenar los datos provenientes del microcontrolador ESP8266 el cual está conectado al sensor MPU6050, esta tabla se denomina “parametrosmpu” en la cual se almacenan datos de aceleración, amplitud, frecuencia, fecha de los registros de cada dato y el identificador único de cada paciente el cual tiene una relación con la tabla “pacientes”.



**Figura 21. Almacenamiento de los datos del sensor MPU6050**

**Elaborado por: investigador**

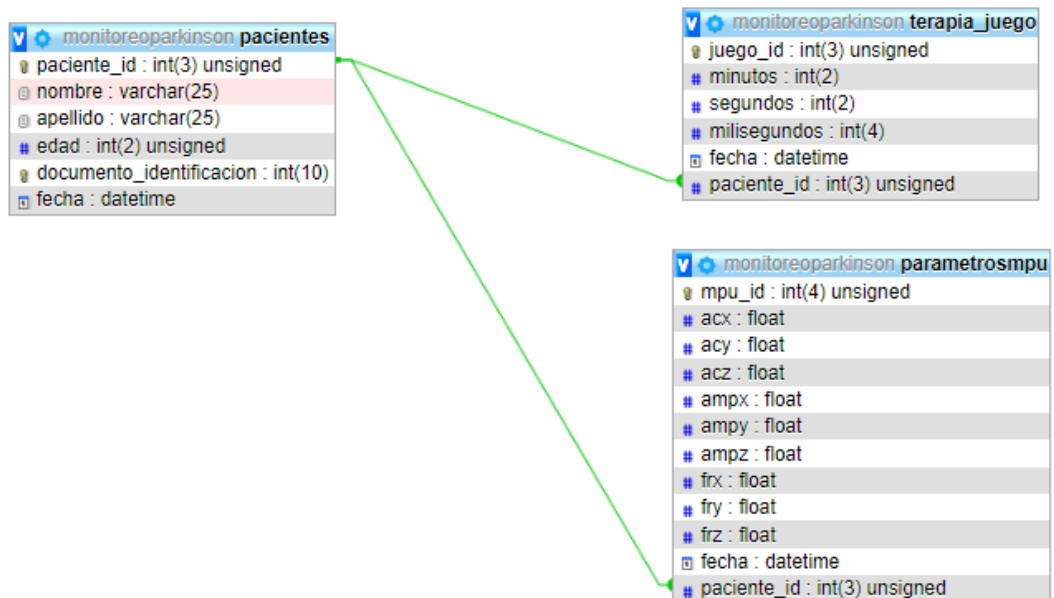
La tercera tabla denominada “terapia\_juego” se utilizó para almacenar los datos de tiempo como minutos, segundos, milisegundos, fecha en que se registró los datos y el identificador del paciente el cual tiene relación con la primera tabla.



**Figura 22. Almacenamiento de los datos de tiempo de la terapia**

Elaborado por: investigador

Para relacionar las tablas se utilizó la función claves foráneas en la base de datos de MariaDB, la cual utiliza una clave primaria llamada normalmente id y la clave foránea se utiliza para relacionar registros únicos en otra tabla. El identificador único de la tabla “pacientes” tiene el nombre de paciente\_id, la cual está relacionada con el campo paciente\_id de la tabla “parametrosmpu” y el campo paciente\_id de la tabla “terapia\_juego”; en la figura 23 se puede observar cómo están relacionadas las tablas.



**Figura 23. Relación de tablas por claves foráneas**

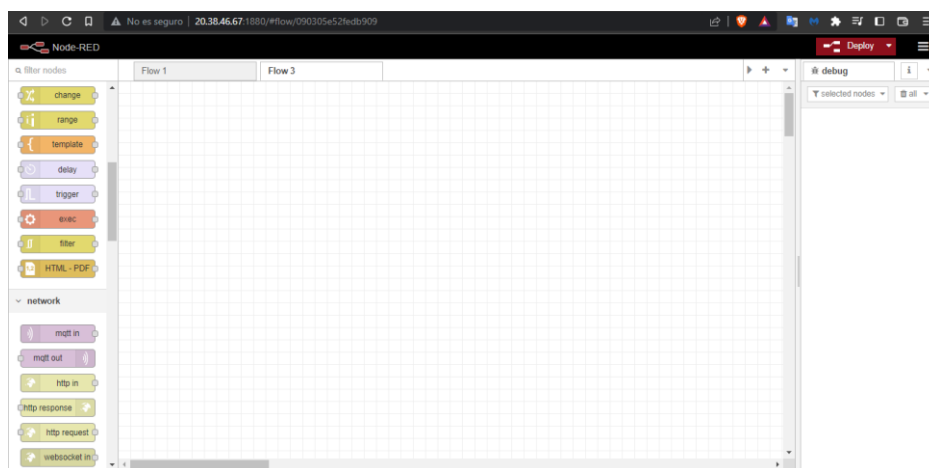
Elaborado por: investigador



## 3.2 Configuración de Node-Red

Node-Red es un software de código abierto, puede ejecutarse en sistemas operativos como Windows, Linux o servidores en la nube, su programación por bloques hace que sea interactivo. Su manejo para comunicar hardware lo hace ideal para aplicaciones IoT, para poder tener una visualización y control en tiempo real para un manejo más simple. Node-Red está basado en JavaScript por lo que permite manejar diferentes aplicaciones y crear diferentes funciones, acorde con la necesidad del usuario. Para ingresar a su interfaz se utiliza la dirección IP del servidor y el puerto por defecto de Node-Red de la siguiente forma `http://20.38.46.67:1880`.

Los valores mostrados en el dashboard de Node-Red provienen desde los nodos sensores, estos valores son mostrados en la interfaz web. Con ellos se observa de mejor forma los valores de amplitud, frecuencia y aceleración en los tres ejes (X, Y, Z).

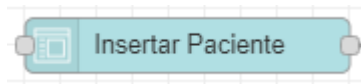


**Figura 24. Interfaz web de Node-Red**

**Elaborado por: investigador**

### 3.2.1 Configuración de Node-Red para el ingreso de datos de los pacientes

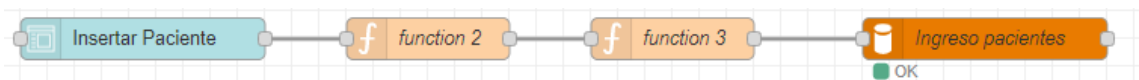
Se realizó el almacenamiento de identificaciones de los pacientes, los valores son ingresados en la base de datos de forma manual los cuales se debe llenar los campos de nombre, apellido, edad y documento de identificación, para realizar esta operación se utilizó el nodo form.



**Figura 25. Nodo Form**

**Elaborado por: investigador**

El nodo de la figura 26, permite el ingreso de los datos de los pacientes, para poder ingresar estos valores en la base de datos se utilizó nodos function y un Node mysql. Para ingresar los datos se utilizó la siguiente sentencia de SQL “INSERT INTO pacientes (nombre, apellido, edad, documento\_identificacion) VALUES (“+ nom + “,” + ap + “,” + age + “,” + ci + “);””. La conexión de estos nodos se muestra en la figura 8.

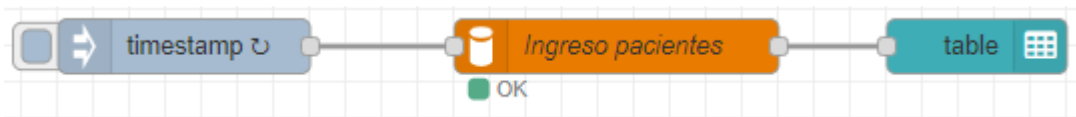


**Figura 26. Ingreso de datos a la base de datos**

**Elaborado por: investigador**

En la figura 27, se muestra la verificación (OK) esto demuestra que está conectado con la base de datos de MariaDB de forma correcta.

Para visualizar todos los pacientes existentes se utilizó el Nodo table en el cual se puede mostrar todos los datos existentes en la base de datos. Para ello se utilizó la sentencia SQL “SELECT \* FROM pacientes;”, esta sentencia está programada en el nodo timestamp el cual se va actualizando cada 30segundos de forma automática.

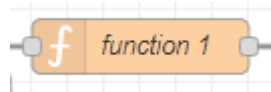


**Figura 27. Nodo table**

**Elaborado por: investigador**

### **3.2.2 Configuración de Node-Red para almacenar los datos del sensor MPU6050**

Para poder conectar a la base de datos, se utilizó un nodo función, el cual está configurado con una sentencia SQL, dicha sentencia toma todos los valores recopilados y los almacena en la base de datos.



**Figura 28. Nodo funtion**

**Elaborado por: investigador**

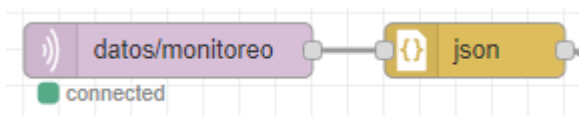
Para poder insertar los valores desde Node-Red hacia la base de datos se utilizó el Nodo mysql el cual permite esta conexión después de una previa configuración en la cual se incluye el nombre de la base de datos, usuario y contraseña.



**Figura 29. Nodo mysql**

**Elaborado por: investigador**

Para obtener los datos que posteriormente serán almacenados se utilizó el Nodo MQTT in, el cual permite obtener todos los datos provenientes de los nodos sensores. Los datos provenientes del sensor (MPU 6050) están en formato json, por lo cual es necesario convertirlos a objetos JavaScript para poder trabajar con ellos. El nodo mqtt in actúa como suscriptor al Bróker de Mosquitto instalado en el servidor Ubuntu Server con el tópico de “datos/monitoreo”.



**Figura 30. Nodo mqtt in y Nodo json**

**Elaborado por: investigador**

La conexión final entre todos los nodos quedó de la siguiente forma, en la cual el nodo function utilizando la secuencia SQL “INSERT INTO `parametrosmpu` (`acx`, `acy`, `acz`, `ampx`, `ampy`, `ampz`, `frx`, `fry`, `frz`, `paciente\_id`) VALUES (" + ax + "," + ay + "," + az + "," + ampx + "," + ampy + "," + ampz + "," + fx + "," + fy + "," + fz + "," + id + ")”;

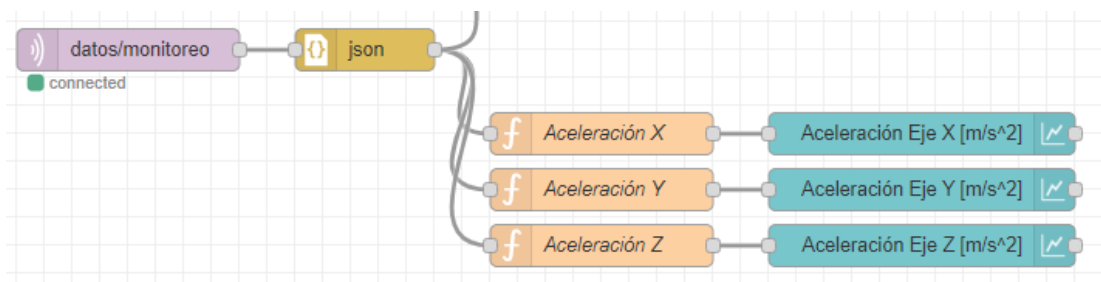
 almacena los datos en la base de datos en la tabla denominada “parametrosmpu”.



**Figura 31. Lectura y almacenamiento de datos**

Elaborado por: investigador

Se utilizó el nodo chart para representar gráficamente todos los datos monitoreados durante el tiempo que el usuario lo utiliza. En la Figura 32 se observa cómo se representa los datos de las aceleraciones en los tres ejes tomadas por el sensor MPU6050.



**Figura 32. Representación gráfica de los datos**

Elaborado por: investigador

### 3.2.3 Configuración de Node-Red para almacenar datos del Dispositivo de terapia

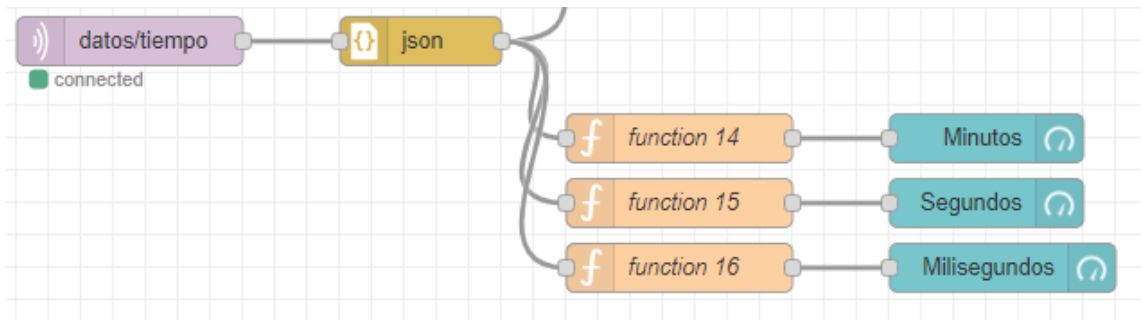
Los datos que se almacenaron en la tabla “terapia\_juego” en la base de datos de MariaDB, se transmiten utilizando el nodo MQTT in el cual recibe los datos de acuerdo con el tópic establecido después es transformado de un objeto Json a un objeto JavaScript posteriormente para por el nodo function el cual contiene una sentencia SQL que inserta los datos utilizando el nodo mysql como se observa en la figura 33.



**Figura 33. Almacenamiento de datos en MariaDB**

Elaborado por: investigador

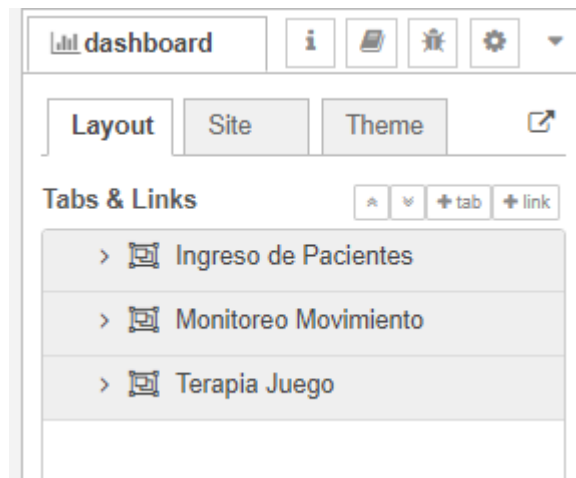
Los datos se representan en gráficas de tipo "gauge" mediante el uso del nodo "gauge". Se utilizan tres nodos para representar los minutos, segundos y milisegundos que se tardaron en realizar la terapia.



**Figura 34. Representación de datos en nodos gauge**

**Elaborado por: investigador**

Para organizar las gráficas, se crearon instancias contenedoras que satisfacen las necesidades individuales del usuario. Estas se muestran en la figura 35, donde se crearon tablas que albergan grupos de nodos gráficos y texto.



**Figura 35. Creación de tablas y grupos**

**Elaborado por: investigador**

El primer tablero “Ingreso de Pacientes” es el que permitió ingresar los datos de los pacientes de forma manual. Los datos ingresados son almacenados en la base de datos de MariaDB. En la figura 36 se puede observar cómo está distribuida los diferentes elementos.



**Figura 36. Interfaz del Dashboard para el ingreso de pacientes**

**Elaborado por: investigador**

En la parte lateral izquierda se realiza la entrada de los datos del paciente a monitorear. En la parte lateral derecha, se pueden ver los datos del último paciente ingresado recientemente. Finalmente, en la parte inferior se encuentra una tabla con los datos de todos los pacientes ingresados.

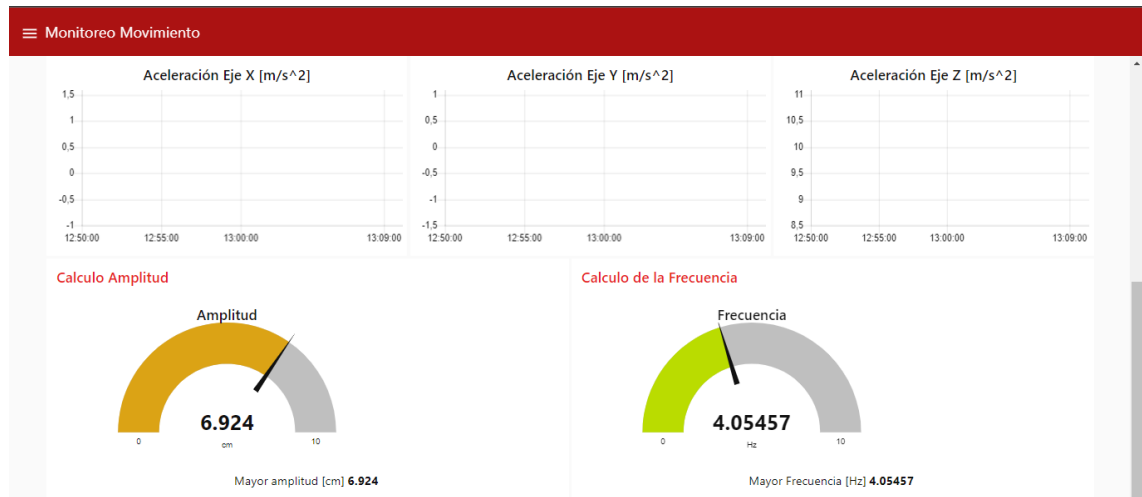
La segunda tabla, llamada "Monitoreo Movimiento", contiene información enviada por el sensor MPU6050. Los datos se representan en gráficas de histograma de los tres ejes. Además, se incluye el valor de la amplitud y la frecuencia máxima calculados durante el período de monitoreo. La figura 37 se puede observar cómo está distribuido en el dashboard.



**Figura 37. Datos del paciente en la Interfaz Monitoreo Movimiento**

**Elaborado por: investigador**

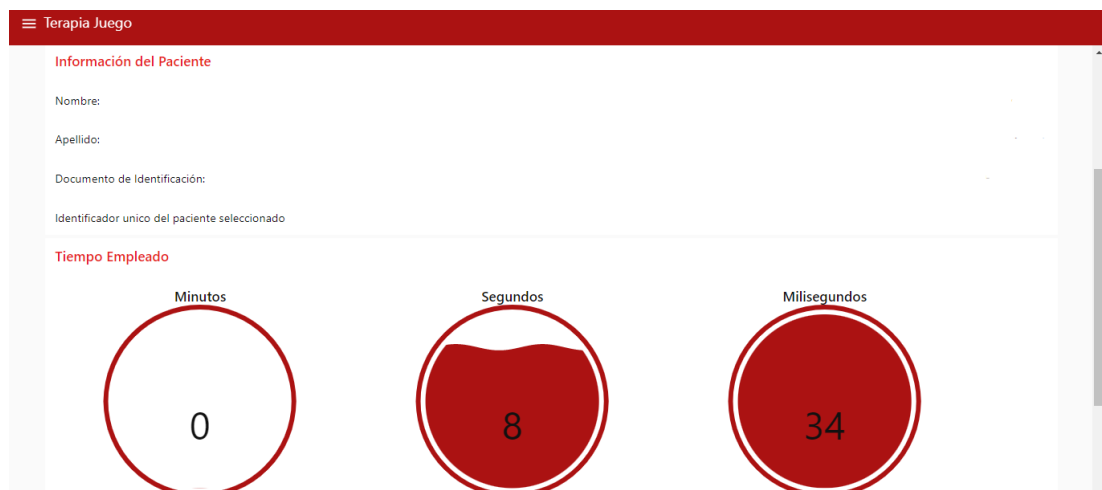
En la figura 38 se presenta la información del paciente que ha sido ingresado para su monitoreo. En dicha figura se puede apreciar cómo se representarán los valores de aceleración en los ejes x, y, z, así como también la amplitud y frecuencia en la interfaz gráfica del dashboard.



**Figura 38. Interfaz gráfica Monitoreo de Movimiento**

**Elaborado por: investigador**

Además, en la tabla "Terapia Juego", la parte superior incluye información del paciente que está realizando la terapia, mientras que la parte inferior cuenta con tres gauges que brindan una representación visual más clara y efectiva para el usuario sobre el tiempo dedicado a la terapia. La figura 39 se puede observar el diseño de la interfaz gráfica del dashboard.



**Figura 39. Interfaz gráfica del dashboard Terapia Juego**

**Elaborado por: investigador**

### **3.3 Flujograma de procesos para el monitoreo con el sensor MPU6050**

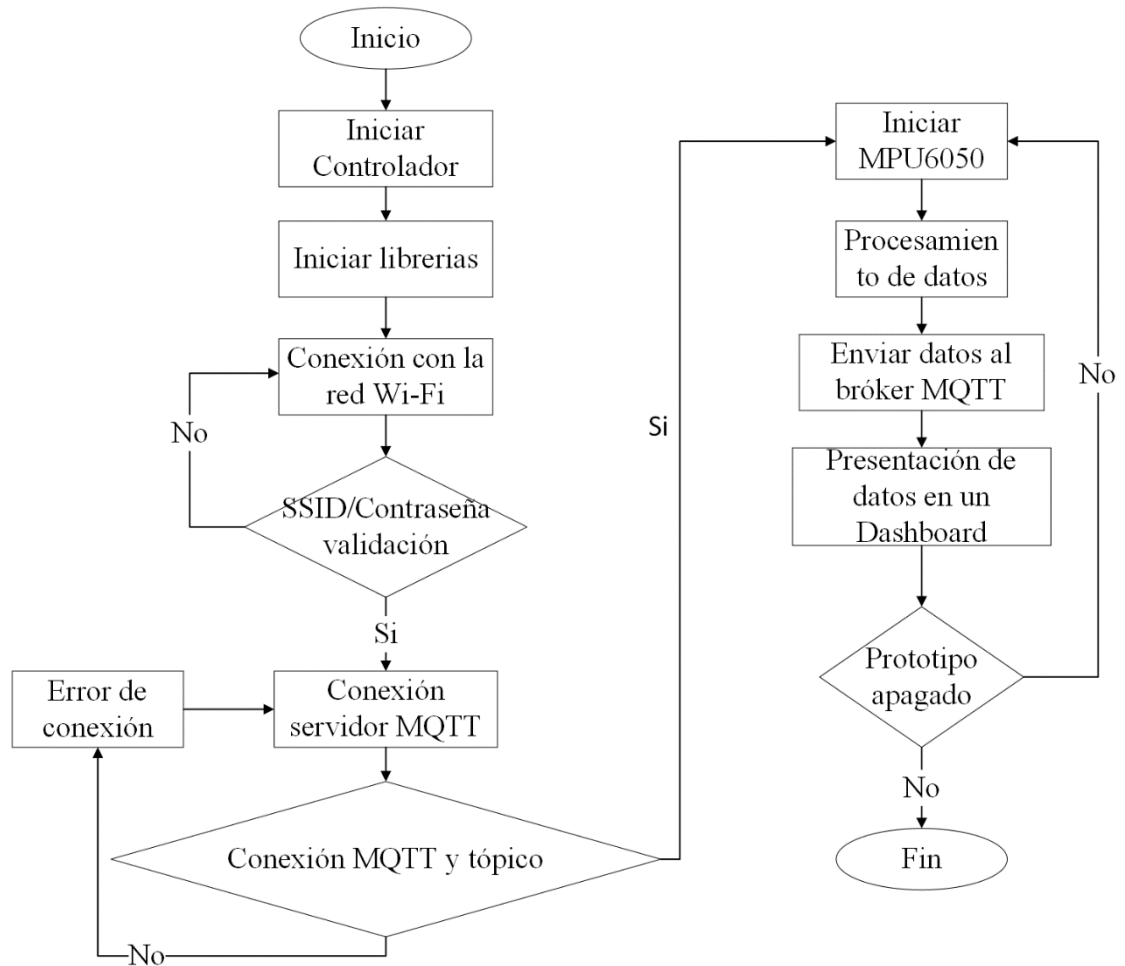
En esta parte se explica el desarrollo de todos los procesos que forman parte del sistema de rehabilitación y monitoreo.

La conexión desde el controlador ESP8266 hacia el servidor de Ubuntu Server en la nube de Azure se describe en la figura 38.

El proceso inicia al encenderse el microcontrolador, el cual tiene cargado un script en Arduino IDE que inicia las librerías. En la siguiente sección se establece la conexión Wi-Fi de la tarjeta ESP8266 hacia la red conectada a internet estableciendo el nombre de la red y la contraseña. Si la conexión fue exitosa, de acuerdo con la programación, se establece la comunicación con el bróker MQTT alojado en el servidor de Ubuntu Server en la nube de Azure y se establece la conexión con el bróker de Mosquitto y el dashboard.

Para que se lleve a cabo las mediciones por parte del sensor MPU6050 la conexión deberá ser exitosa. Los datos obtenidos del sensor toman parte el proceso denominado procesamiento de datos, en el cual la ESP8266 se encarga de procesar la información para ser enviados al Bróker mediante la conexión Wi-Fi. El Bróker recibe todos los mensajes publicados por los usuarios y los distribuye entre ellos. En la siguiente sección, los datos recibidos son mostrados en una interfaz gráfica en donde se puede visualizar la amplitud, aceleración en cada uno de los ejes y la frecuencia. Todo este proceso se realiza cuando el dispositivo se encuentre encendido.





**Figura 40. Diagrama de flujo monitoreo sensor MPU6050**

Elaborado por: investigador

### 3.4 Adquisición de datos con el Sensor MPU6050

Para la adquisición de los datos se ocupó el sensor MPU6050 en cual es un sensor que utiliza la comunicación I2C utilizando los pines SCL y SDA de la tarjeta ESP8266 los datos obtenidos por parte del sensor son la aceleración en los tres ejes X, Y y Z. Las librerías utilizadas son para la conexión del sensor, funcionamiento del sensor y establecer conexión por I2C, estas se muestran en la figura 41.

```

#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"
  
```

**Figura 41. Librerías de Arduino para el sensor MPU6050**

Elaborado por: investigador

### 3.4.1 Cálculo de la amplitud de los temblores

A partir de los datos obtenidos de aceleración que proporciona el sensor MPU6050 se puede calcular el desplazamiento de un objeto en un tiempo dado, esto se hace utilizando la siguiente ecuación[34]:

$$S = S_o + \frac{1}{2}at^2 \quad (1)$$

donde:

$S$  = desplazamiento

$S_o$  = desplazamiento inicial

$a$  = aceleración

$t$  = tiempo

Esta fórmula es aplicada para los valores de aceleración de los ejes X, Y y Z, obtenidos del sensor MPU6050, el algoritmo de programación calcula las amplitudes en centímetros para cada eje utilizando las siguientes ecuaciones:

$$S_X = S_{Xo} + \frac{1}{2}a_X T_s^2 \quad (2)$$

$$S_Y = S_{Yo} + \frac{1}{2}a_Y T_s^2 \quad (3)$$

$$S_Z = S_{Zo} + \frac{1}{2}a_Z T_s^2 \quad (4)$$

donde:

$S_X, S_Y$  y  $S_Z$  = desplazamiento de cada eje en cm.

$S_{Xo}, S_{Yo}$  y  $S_{Zo}$  = valores iniciales de cada desplazamiento en cm.

$a_X, a_Y$  y  $a_Z$  = aceleración de cada eje en  $m/s^2$ .

$T_s$  = periodo entre cada una de las muestras de aceleración.

El algoritmo de programación calcula las amplitudes en cada cambio de dirección durante el tiempo que dura la prueba y al final entrega la mayor amplitud que se calculó. En la figura 42 se puede observar las líneas de código que permiten obtener las aceleraciones en los tres ejes mediante un comando de la librería de Arduino para

el sensor MPU6050 “mpu.getAcceleration(&ax, &ay, &az)”, comando que permite obtener en bits las aceleraciones medidas por el sensor, mismas que requieren una conversión a  $m/s^2$ . Finalmente se realiza el cálculo de las amplitudes para los tres ejes en donde no se considera  $S_{Xo}$ ,  $S_{Yo}$  y  $S_{Zo}$  ya que parten desde el reposo y su desplazamiento inicial es 0.

```
void aceleraciones_amplitudes(){
  // Leer las aceleraciones
  mpu.getAcceleration(&ax, &ay, &az);
  acx = ax * accScale;
  acy = ay * accScale;
  acz = az * accScale;

  //Calcular amplitud
  sx = 50 * acx * (Ts*Ts);
  sy = 50 * acy * (Ts*Ts);
  sz = 50 * acz * (Ts*Ts);
}
```

**Figura 42. Cálculo de aceleraciones y amplitudes**

**Elaborado por: investigador**

### 3.4.2 Cálculo de las Frecuencias de los Temblores

Para obtener la frecuencia de los temblores se utilizó la Transformada Rápida de Fourier (FFT). El algoritmo de programación en Arduino utiliza la librería “arduinoFFT.h” que permite calcular las frecuencias para cada una de las amplitudes obtenidas de los ejes X, Y y Z de los valores de aceleración proporcionados por el sensor MPU6050.

```
const uint16_t samples = 64;
const double signalFrequency = 4;
const double samplingFrequency = 100;
```

**Figura 43. Variables para el cálculo de la FFT**

**Elaborado por: investigador**

En la figura 44 se puede observar los valores de la señal de frecuencia y el valor de la frecuencia de muestro, al igual que el número de muestras que esta tomará; estos parámetros son necesarios ya que son utilizados por la librería “arduinoFFT.h”. Para calcular la frecuencia estos valores son definidos en base al teorema de Nyquist, en el cual la frecuencia de muestro debe ser algo mayor o el doble de la señal de frecuencia media [35].

El número de muestras según la librería puede ser cualquier número que se obtenga a partir de calcular un número a una potencia de dos.

$$samples = 2^6 = 64$$

```
void frecuenciaX(){
  //Calculo FFT
  double amplitud = sx;
  /* Build raw data */
  double cycles = (((samples-1) * signalFrequency) / samplingFrequency); //Number of signal cycles that the sampling will read
  for (uint16_t i = 0; i < samples; i++)
  {
    vReal[i] = int8_t((amplitud * (sin((i * (twoPi * cycles)) / samples))) / 2.0); /* Build data with positive and negative values*/
    vImag[i] = 0.0;
  }
  /* Print the results of the simulated sampling according to time */
  FFT.Windowing(vReal, samples, FFT_WIN_TYP_HAMMING, FFT_FORWARD); /* Weigh data */
  FFT.Compute(vReal, vImag, samples, FFT_FORWARD); /* Compute FFT */
  FFT.ComplexToMagnitude(vReal, vImag, samples); /* Compute magnitudes */
  fx = FFT.MajorPeak(vReal, samples, samplingFrequency);
  if(isnan(fx)){
    fx=0;
  }
}
```

**Figura 44. Código del cálculo de la frecuencia en Arduino**

**Elaborado por: investigador**

También en la figura 44 se observa cómo se calcula la frecuencia de la amplitud del eje X. La función “FFT.MajorPeak” calcula el punto de mayor de frecuencia de las 64 muestras tomadas y ese es el valor de la FFT que el programa entrega, cuando el valor de la amplitud es cercano a cero el valor de frecuencia es nulo por lo que se agregó una condición que cuando ocurra esos casos la frecuencia sea cero.

La frecuencia es calculada para cada una de las amplitudes obtenidas de los ejes X, Y y Z y son enviadas a la base de datos.

Para la transmisión de los datos de forma inalámbrica por medio de una red Wi-Fi se utilizó una librería y para transmitir los datos mediante el protocolo MQTT se utilizó dos librerías ya que los datos se envían en formato Json hacia el bróker MQTT.

```
#include <WiFiManager.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
```

**Figura 45. Librerías para la transmisión de datos**

**Elaborado por: investigador**

### 3.5 Diseño esquemático

#### 3.5.1 Esquema de conexión del dispositivo de monitoreo

En la figura 46 se presenta una representación gráfica de cómo están conectados los componentes electrónicos seleccionados previamente para el monitoreo de señales en pacientes con EP. Este diagrama de conexiones fue diseñado utilizando el software de simulación Fritzing.

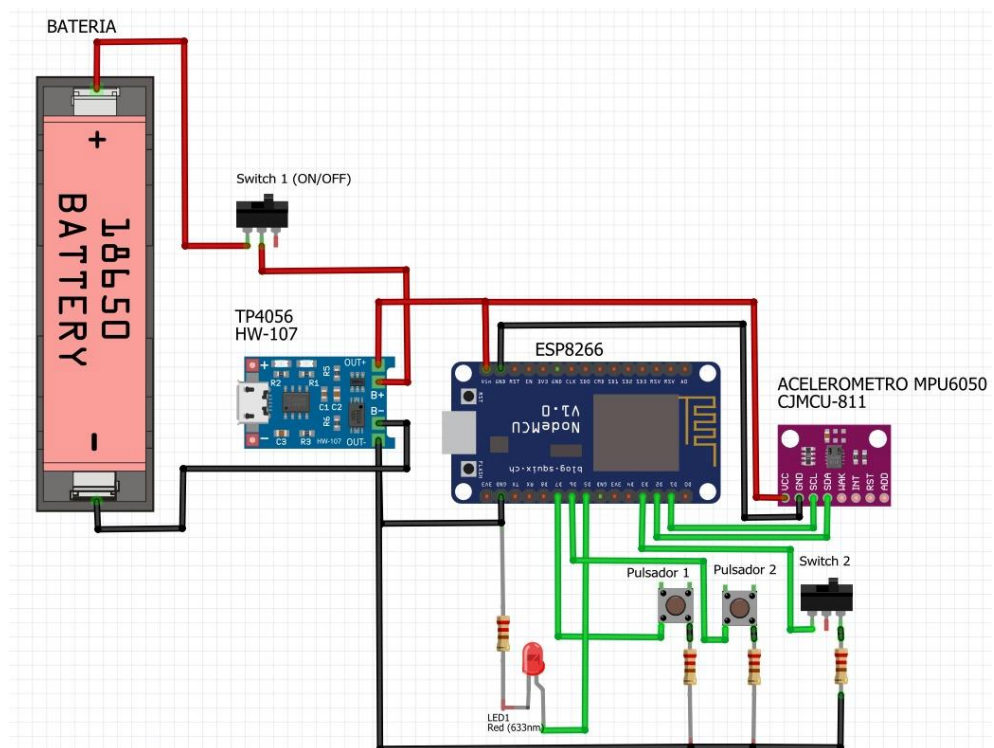


Figura 46. Esquema de conexión de los componentes electrónicos del dispositivo sensor de monitoreo de temblores para EP

Elaborado por: investigador

#### 3.5.2 Esquema de conexión del circuito para terapias de pacientes con EP

De igual forma, con la ayuda del software de simulación Fritzing se diseñó el esquema de conexión del sistema que ayuda a la rehabilitación convencional de los pacientes con la EP; en la figura 47 se muestra el diagrama de conexión entre la ESP8266, la fuente de alimentación LCD 16X2 y los finales de carrera.

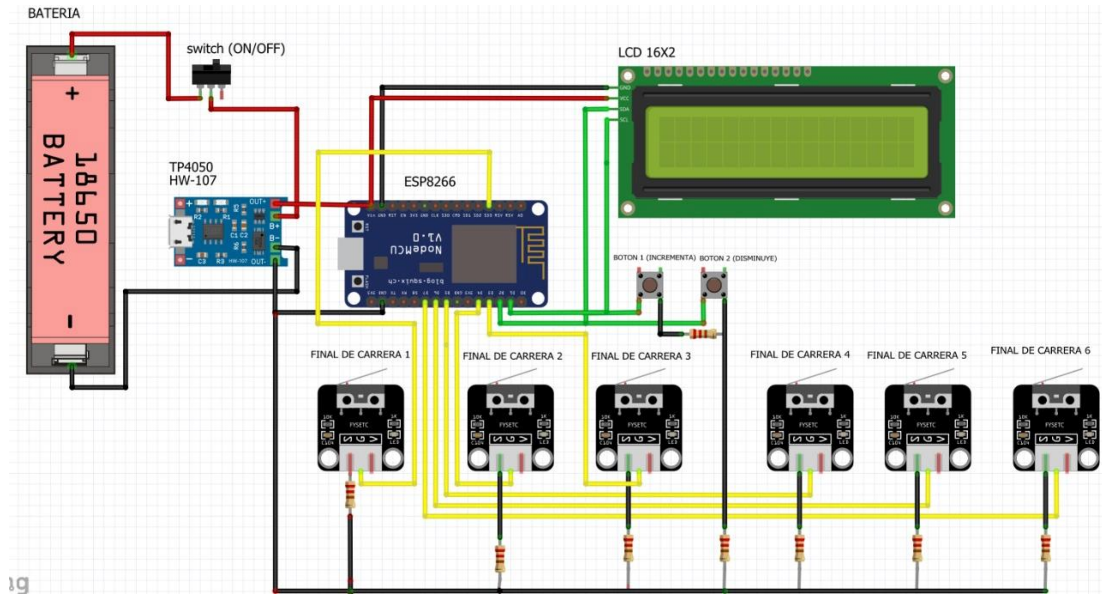


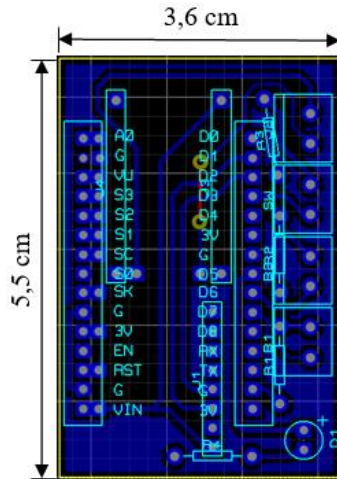
Figura 47. Esquema de conexión de los componentes electrónicos del dispositivo de terapia

Elaborado por: investigador

### 3.6 Diseño del circuito en PCB

#### 3.6.1 Dispositivo monitoreo de temblores para EP

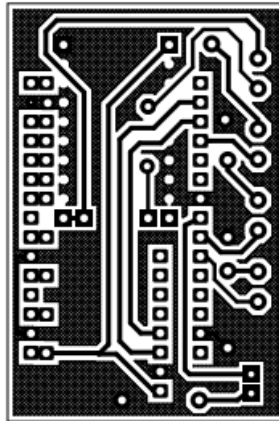
Para poder fabricar la PCB (Print Circuit Board) del circuito de monitoreo de señales, se tuvo como punto de partida el esquemático de la figura 45 y mediante software de simulación Proteus se estableció la ubicación de los componentes y las respectivas conexiones, tomando en cuenta la dimensión de la placa y la anchura de los planos conductores. De este modo en la figura 48, se puede observar el diseño de las pistas del circuito para el monitoreo de señales.



**Figura 48. Diseño del PCB de monitoreo**

**Elaborado por: investigador**

El diagrama final de pistas se imprimió en papel transfer y se puede observar en la figura 49.



**Figura 49. Diagrama de pistas de la PCB diseñada**

**Elaborado por: investigador**

Para la elaboración de la placa electrónica se empleó la técnica o método del planchado. Se soldaron todos los elementos electrónicos en la PCB terminando de esta manera la construcción del dispositivo sensor de monitoreo de temblores para EP. En la figura 50, se representa el circuito implementado, donde se especifica el uso que tienen destinado las borneras.

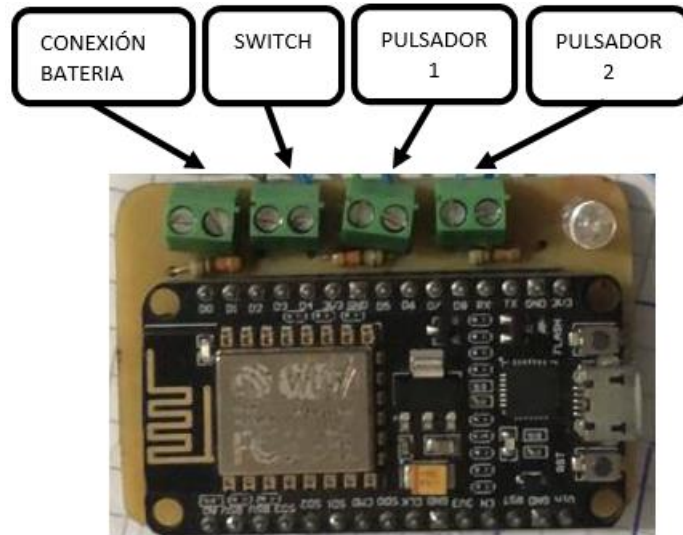


Figura 50. Implementación del circuito final

Elaborado por: investigador

### 3.6.2 Dispositivo de terapia

Se realizó la PCB (Print Circuit Board) a partir del esquemático de conexión del circuito para terapias de pacientes con EP, se empleó el software de simulación Proteus para conectar cada uno de los componentes mediante pistas conductoras. En la figura 51, se puede observar el diagrama de pistas.

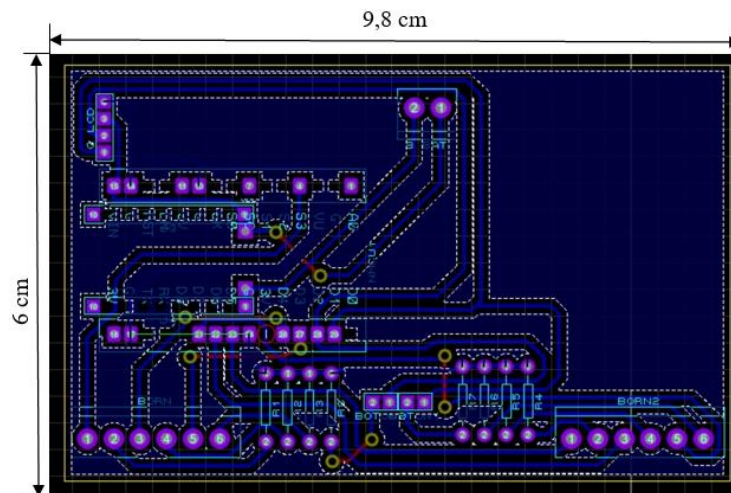
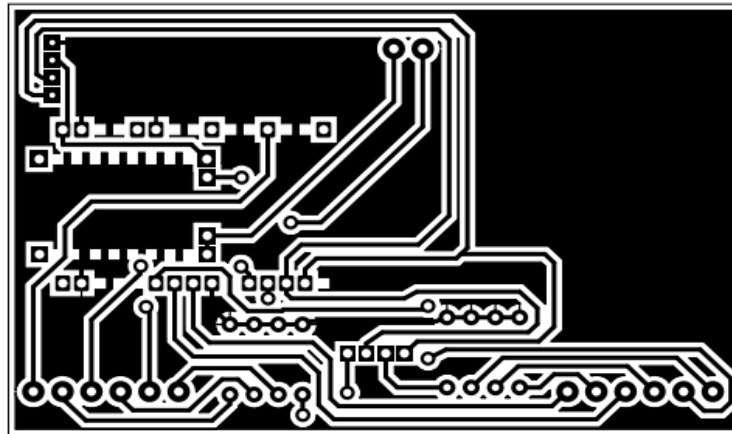


Figura 51. Diseño del PCB de dispositivo de terapia

Elaborado por: investigador



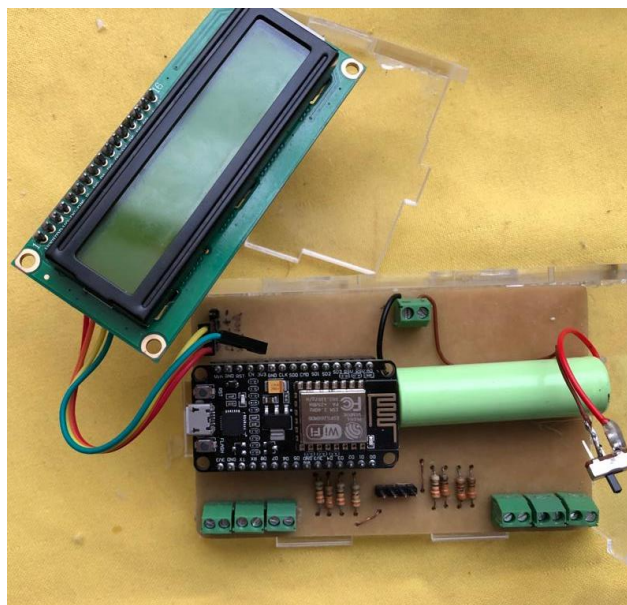
El diagrama de pistas final se imprimió en papel transfer y se puede observar en la figura 52.



**Figura 52. Diagrama de pistas de la PCB diseñada**

**Elaborado por: investigador**

Se realizó las pistas a través de método del planchado y se soldó de forma manual los componentes electrónicos en la PCB. Con esto, se culminó la elaboración del circuito para terapias de los pacientes con EP. En la figura 53 se muestra los dispositivos (LCD 16X2, batería, switch de encendido) conectados a la placa principal.

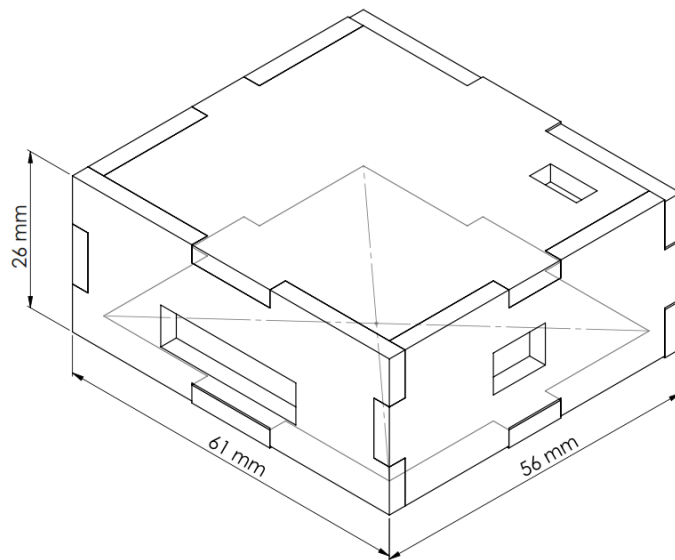


**Figura 53. Implementación del dispositivo de terapia**

**Elaborado por: investigador**

### 3.7 Diseño del case

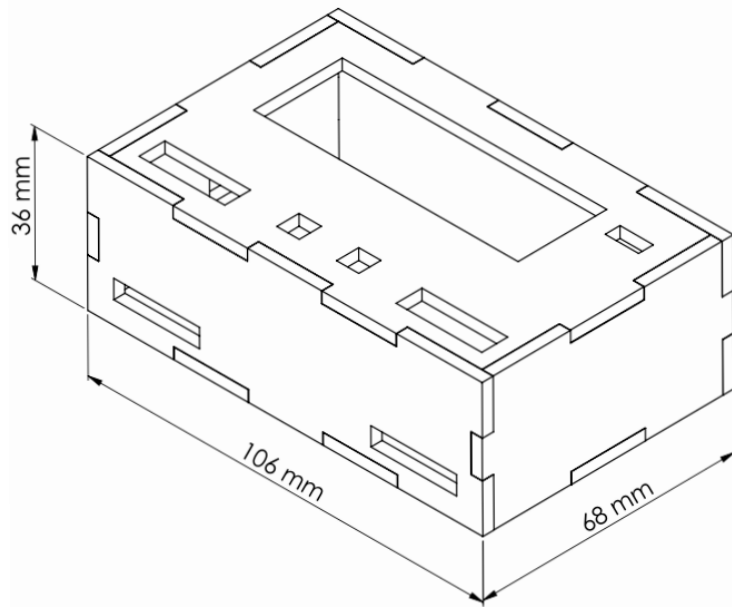
Luego de verificar las conexiones del prototipo, la programación, el tamaño de la placa, las dimensiones y ubicación de los pulsadores e interruptores y fuente de carga, se diseñó una estructura para colocar la placa y sus componentes electrónicos y así el sistema de monitoreo sea portable. La caja se diseñó en el software SolidWorks, cada superficie de la caja se realizó individualmente y luego se unió teniendo así una perspectiva general del diseño. En la figura 54 se muestra el diseño en 3D, mientras que en el anexo A se muestran las medidas de cada una de las caras de la caja.



**Figura 54. Case en 3D para placa de monitoreo**

**Elaborado por: investigador**

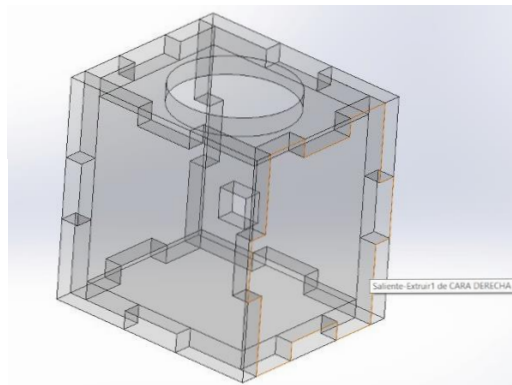
Al igual que el case para la placa de monitoreo, se diseñó una caja que contenga la placa del circuito de terapia. En la figura 55 se puede observar el diseño en 3D, mientras que en el anexo B se muestran las medidas de cada una de las caras de la caja.



**Figura 55. Case en 3D placa del dispositivo de terapia**

**Elaborado por: investigador**

Adicional a la caja principal, se diseñó 6 cajas pequeñas para que contengan los finales de carrera conectados a la placa del dispositivo de terapia. En la figura 56 se puede observar la caja en 3D diseñada en Solid Works.

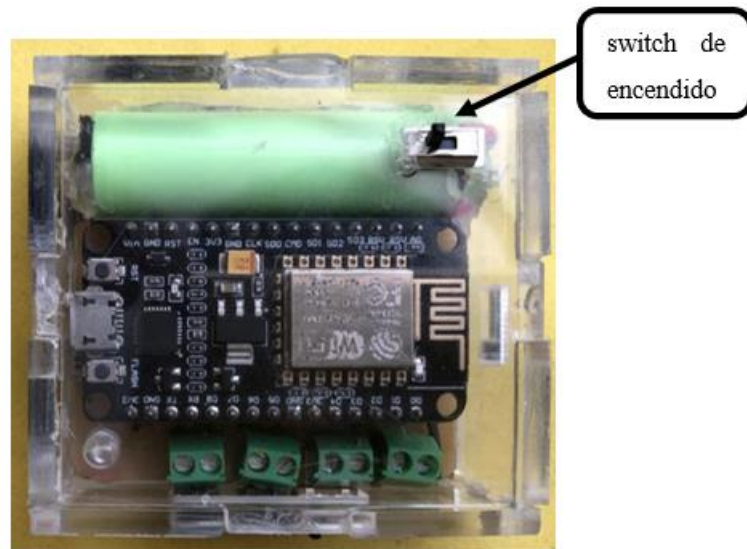


**Figura 56. Case para insertar los objetos esféricos**

**Elaborado por: investigador**

Al culminar con el diseño de las cajas en SolidWorks, se generó el archivo dwg (DraWinG), siendo este un formato de archivo informático de dibujo computarizado, el mismo que permite el corte del acrílico a láser. En la figura 57 se puede observar el

dispositivo de monitoreo dentro de la carcasa, mientras que en la figura 58 se puede observar el dispositivo de terapias dentro de la caja diseñada.



**Figura 57. Colocación de la placa de monitoreo en el case**

**Elaborado por: investigador**



**Figura 58. Colocación de la placa de terapias en el case**

**Elaborado por: investigador**

Para fabricar las carcasas de los finales de carrera, se realizó el corte en madera tipo MDF de 0,3 cm de grosor. En la figura 59 se puede observar los finales de carrera colocados en cada una de las cajas.

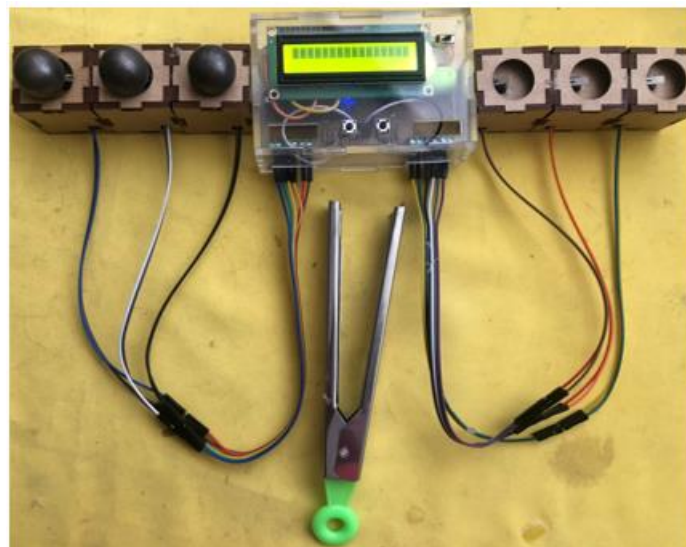


**Figura 59. Cajas contenedoras de objetos circulares para la terapia**

**Elaborado por: investigador**

### **3.8 Dispositivo de terapia convencional**

Adicional al sistema de monitoreo de señales se diseñó un dispositivo donde se pueda realizar una actividad de terapia ocupacional, el cual consiste en atrapar y transportar una esfera con una pinza. Este ejercicio permite a los pacientes con EP trabajar la fuerza en los dedos y mano, y al mismo tiempo le da estabilidad en el codo y el hombro. En la figura 60 se puede observar el dispositivo de terapia.



**Figura 60. Dispositivo para actividad de terapia convencional**

**Elaborado por: investigador**

### 3.9 Soporte para el dispositivo sensor de temblores para EP

Una vez elaborada la PCB y su respectiva carcasa, se elaboró un soporte para colocar el dispositivo, el mismo que se puede observar en la figura 61. El soporte está compuesto de un guante ortopédico y velcro colocado en la parte superior del guante. También se pegó velcro en la cara inferior (piso) del dispositivo de monitoreo para que se una al soporte. El dispositivo completo sobre el soporte se puede observar en la figura 62.



**Figura 61. Soporte para el dispositivo de monitoreo**

**Elaborado por: investigador**



**Figura 62. Dispositivo de monitoreo de temblores para EP en el soporte para mano**

**Elaborado por: investigador**

### 3.10 Pruebas de funcionamiento

#### 3.10.1 Vinculación del Dispositivo Electrónico de Monitoreo de Temblores con la red inalámbrica

Para empezar a realizar la evaluación, en primer lugar, se debe colocar el Dispositivo Electrónico de Monitoreo en la mano del paciente y accionar el interruptor de encendido. Inmediatamente después, se debe conectar a una red Wi-Fi inalámbrica mediante un teléfono móvil o Tablet. En la figura 63 se puede observar la secuencia para conectar el Dispositivo Electrónico de Monitoreo a una red inalámbrica.



Figura 63. Conexión del sistema de monitoreo a la red local

Elaborado por: investigador

Una vez establecida la conexión, el siguiente paso es ingresar a la Interfaz Gráfica Monitoreo de Movimiento, donde se ingresa los datos del paciente o en el caso de que ya este registrado el paciente solo seleccionamos e iniciamos el monitoreo de los temblores del paciente con EP.

#### 3.10.2 Test de evaluación

Para el monitoreo de la intensidad de los temblores de los pacientes con EP se realizó tres pruebas diferentes, TEST 1, TEST 2 y TEST 3. Estos TESTs están basados en los tipos de temblores que aparecen en las extremidades superiores mientras el paciente realiza una acción. A continuación, se describirá cada tipo de prueba y la forma en que deben ser de realizadas.

### **TEST 1 (Temblor postural de las manos)**

El paciente adopta una postura sostenida contra la fuerza de gravedad, pero sin realizar ningún movimiento. En este caso se estira el brazo hacia delante con las palmas de las manos hacia abajo. La muñeca debe estar recta y los dedos cómodamente separados de tal forma que no se toquen entre sí [36].

La postura que debe tomar el paciente para la evaluación de temblor postural de las manos se muestra a continuación en la figura 64.



**Figura 64. Postura para evaluar temblor postural de las manos**

**Elaborado por: investigador**

### **TEST 2 (Temblor cinético o de acción)**

Se da cuando la persona quiere realizar un movimiento de forma consciente y voluntaria. En esta prueba se evalúa con la maniobra dedo-nariz, partiendo de la posición con los brazos estirados el paciente debe tocar 10 veces la punta de la nariz. La maniobra dedo-nariz debe ejecutarse lo suficientemente lenta para que no se encubra cualquier temblor; se debe repetir el ejercicio con la otra mano para evaluar por separado el comportamiento de cada mano.

La postura que debe tomar el paciente para la evaluación de temblor cinético de las manos se puede observar en la figura 65.





(a)

(b)

**Figura 65. Ejercicio para la evaluación de Temblor de Acción de las Manos. El ejercicio comienza con la postura (a), después lentamente se pasa a la postura (b) y culmina al adoptar nuevamente la postura (a)**

**Elaborado por: investigador**

### **TEST 3 (Amplitud del temblor en reposo)**

La persona se encuentra recostada o sentada apoyando la parte del cuerpo en la que se da el temblor. En este caso el paciente debe estar sentado tranquilamente en una silla con las manos apoyadas en el reposa- brazos y los pies de forma cómoda, durante 20 segundos, sin recibir ninguna otra indicación. El temblor de reposo se evalúa por separado para cada extremidad.

La postura que debe tomar el paciente para la evaluación de temblor en reposo de las manos se puede observar en la figura 66.



**Figura 66. Postura para evaluación Amplitud Temblor en Reposo**

**Elaborado por: investigador**

### **3.10.3 Resultados de las evaluaciones**

En este punto se presenta los resultados de las evaluaciones con las cuales se comprobó el funcionamiento del sistema. Para ello se evaluaron a 2 pacientes diagnosticados con EP y 2 pacientes sanos.

Se realizaron evaluaciones en pacientes cuyas identidades se mantienen en oculto debido a la privacidad requerida; sin embargo, se incluyen ciertos datos, como género, edad y un identificador único, como se muestra en la tabla 10.

**Tabla 10. Datos de pacientes evaluados**

	<b>Identificador único</b>	<b>Nombre</b>	<b>Edad</b>	<b>Genero</b>
Pacientes sanos	1	NN1	26	Femenino
	2	NN2	56	Masculino
Paciente con EP	3	NN3	49	Masculino
	4	NN4	78	Femenino

**Elaborado por: investigador**

En la figura 67 se pueden observar los pacientes y cómo se realizaron las pruebas de funcionamiento del dispositivo de monitoreo y rehabilitación para pacientes con EP.



**Figura 67. Pruebas de funcionamiento en Pacientes Sanos y con EP**

**Elaborado por: investigador**

Se evaluó a todos los pacientes siguiendo las instrucciones para cada tipo de prueba (TEST 1, TEST 2, TEST 3) detalladas en la sección 3.10.2.

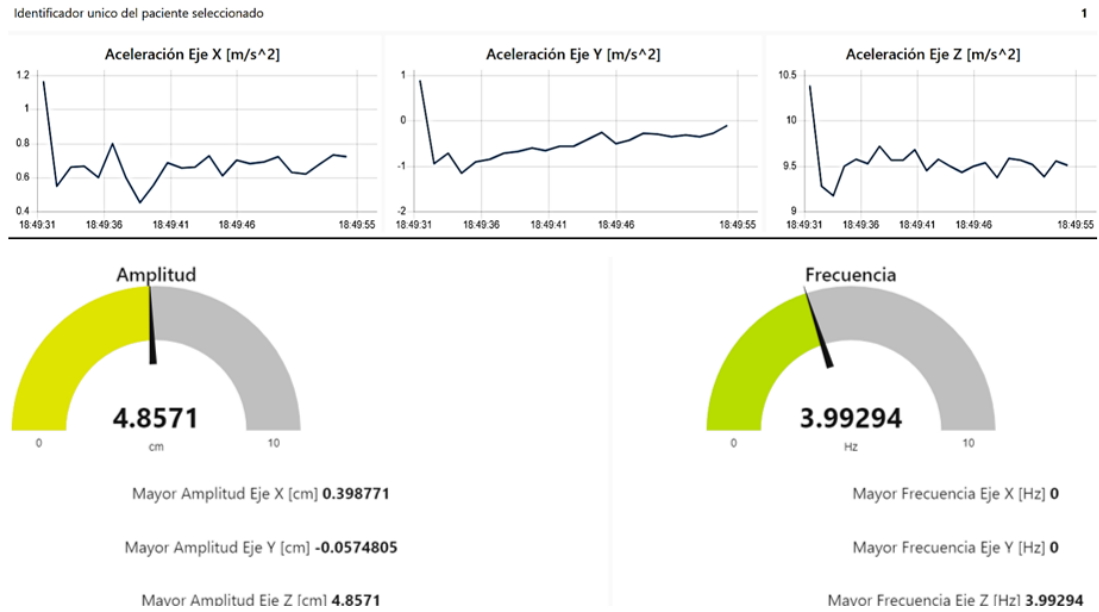
Primero se presentan los resultados de las evaluaciones realizadas a los pacientes sanos y, a continuación, se presentan los resultados de los pacientes con enfermedad de Parkinson.

#### ***3.10.4 Resultados de las evaluaciones realizadas a los pacientes sanos***

Debido a la similitud de resultados de la evaluación de los pacientes sanos, en esta sección se incluirá únicamente las imágenes que muestran los resultados de los TESTs realizados al paciente sano NN1. Sin embargo, se realizó todas las tablas correspondientes a los resultados de los tres TESTs realizados a cada paciente sano.

## Resultado de la evaluación realizada al paciente sano NN1

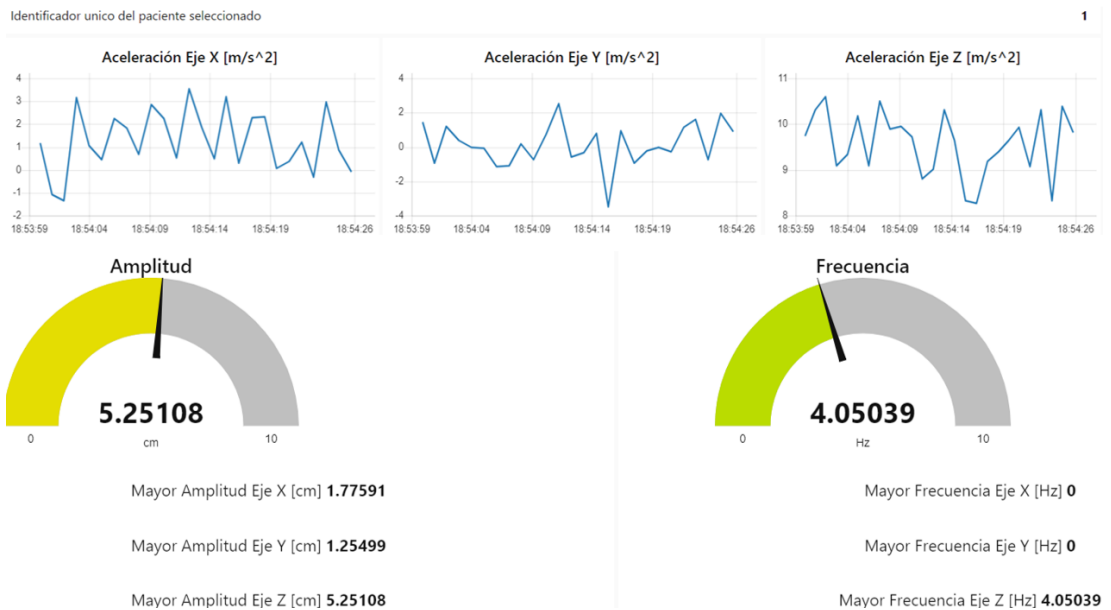
En la figura 68 se muestra los resultados del TEST 1 realizado al paciente sano NN1.



**Figura 68. Resultado del TEST 1 realizado al paciente sano NN1**

Elaborado por: investigador

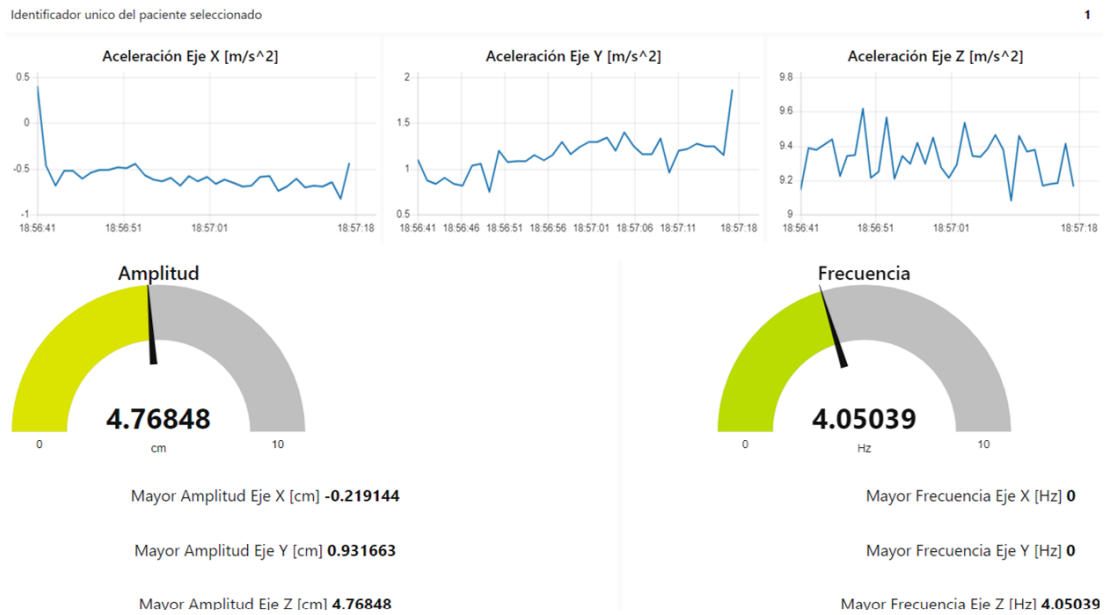
En la figura 69 se muestra los resultados del TEST 2 realizado al paciente sano NN1.



**Figura 69. Resultado del TEST 2 realizado al paciente sano NN1**

Elaborado por: investigador

En la figura 70 se representan los resultados del TEST 3 realizado al paciente sano NN1.



**Figura 70. Resultado del TEST 3 realizado al paciente sano NN1**

**Elaborado por: investigador**

Los resultados obtenidos en cada TEST realizado al paciente sano NN1 se puede observar en la tabla 11.

**Tabla 11. Resultado de la evaluación realizada al paciente sano NN1**

PACIENTE SANO NN1								
Identificador	AMPLITUD MAXIMA				FRECUENCIA MAXIMA			
	Eje	TEST 1	TEST 2	TEST 3	Eje	TEST 1	TEST 2	TEST 3
1	X[cm]	0,39	1,77	-0,21	X [Hz]	0	0	0
	Y [cm]	-0,05	1,25	0,93	Y [Hz]	0	0	0
	Z [cm]	4,85	5,25	4,76	Z [Hz]	3,992	4,05	4,05

**Elaborado por: investigador**

## Resultado de la evaluación realizada al paciente sano NN2

Los resultados obtenidos en cada TEST realizado al paciente sano NN2 se puede observar en la tabla 12.

Tabla 12. Resultado de la evaluación realizada al paciente sano NN2

PACIENTE SANO NN2								
Identificador	AMPLITUD MAXIMA				FRECUENCIA MAXIMA			
	2	Eje	TEST 1	TEST 2	TEST 3	Eje	TEST 1	TEST 2
	X[cm]	0,38	2,58	0,37	X [Hz]	0	4,01	0
	Y [cm]	-0,002	0,33	-0,01	Y [Hz]	0	0	0
	Z [cm]	4,84	5,63	4,84	Z [Hz]	3,99	4,04	3,98

Elaborado por: investigador

### Análisis de los resultados de las evaluaciones realizadas a los pacientes sanos

Como era de esperar, los pacientes sanos sin síntomas de desorden neurológico presentaron resultados esperados en el TEST 1. Los datos obtenidos de amplitud son bajos y no existe una mayor variación en sus lecturas, ya que son cercanas a 0. De igual manera, su frecuencia es cero en los ejes X y Y, a diferencia del eje Z, que muestra una frecuencia de 4 Hz debido a que el eje Z en reposo tiene una aceleración constante cercana a la de la gravedad de la tierra de  $9,8 \text{ m/s}^2$ .

En el TEST 2 se observa que los datos obtenidos de amplitud de los pacientes sin EP existen variación en sus lecturas ya que en este test se realizan movimientos, por lo que los valores de aceleración se incrementan, mientras que su frecuencia es cero en los ejes X y Y a diferencia del eje Z que muestra una frecuencia de 4,05 y 3,9 Hz debido a que el eje Z tiene una aceleración constante cercana a la de la gravedad de la tierra de  $9,8 \text{ m/s}^2$ .

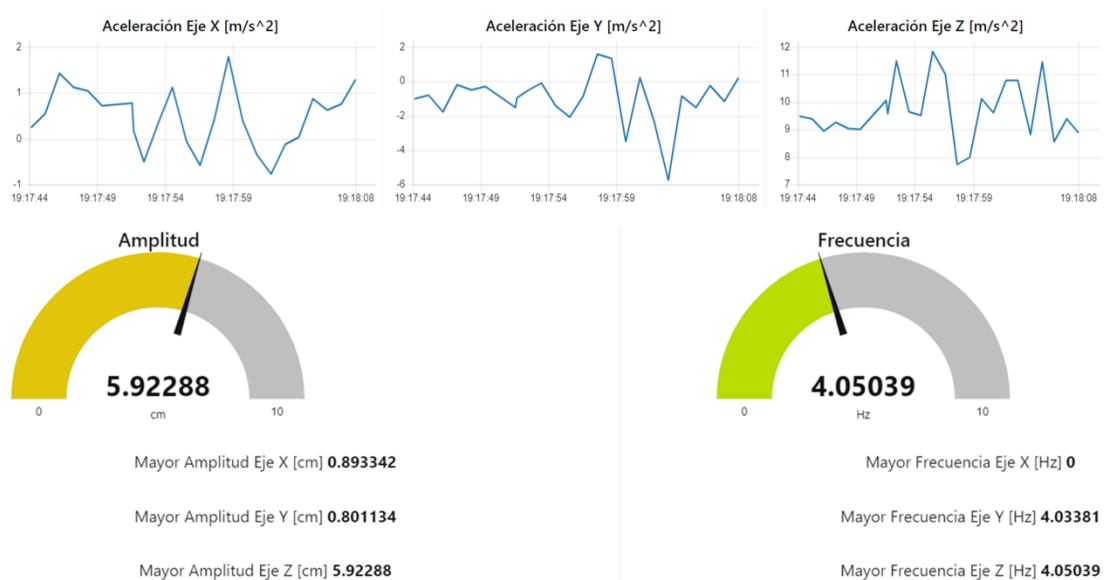
En el Test 3, los datos obtenidos de amplitud del paciente sin EP presentan variación en sus lecturas ya que en este test la posición del brazo provoca que exista una variación en los ejes X y Y, por lo cual los valores de aceleración incrementan, mientras que la frecuencia es cero en los ejes X y Y a diferencia del eje Z que muestra una frecuencia de 4,05 y 3,98 Hz debido a que el eje Z tiene una aceleración constante cercana a la de la gravedad de la tierra de  $9,8 \text{ m/s}^2$ .

### 3.10.5 Resultados de las evaluaciones realizadas a los pacientes con EP

En esta sección se incluirán todas las capturas tomadas del dashboard que muestran los resultados de los TESTs realizados a cada uno de los pacientes con EP. Con ello se podrá visualizar gráficamente el comportamiento de los temblores en cada uno de estos pacientes.

#### Resultados de la evaluación realizada al paciente NN3 con EP

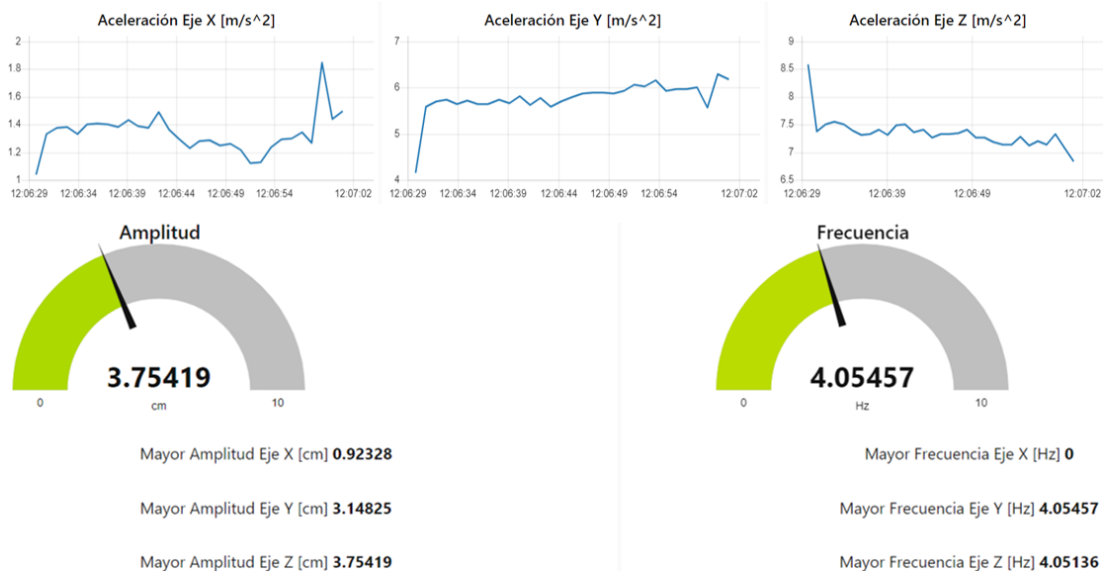
En la figura 71 se representan los resultados del TEST 1 realizado al paciente NN 3 con EP.



**Figura 71. Resultados del TEST 1 del paciente NN3 con EP.**

**Elaborado por: investigador**

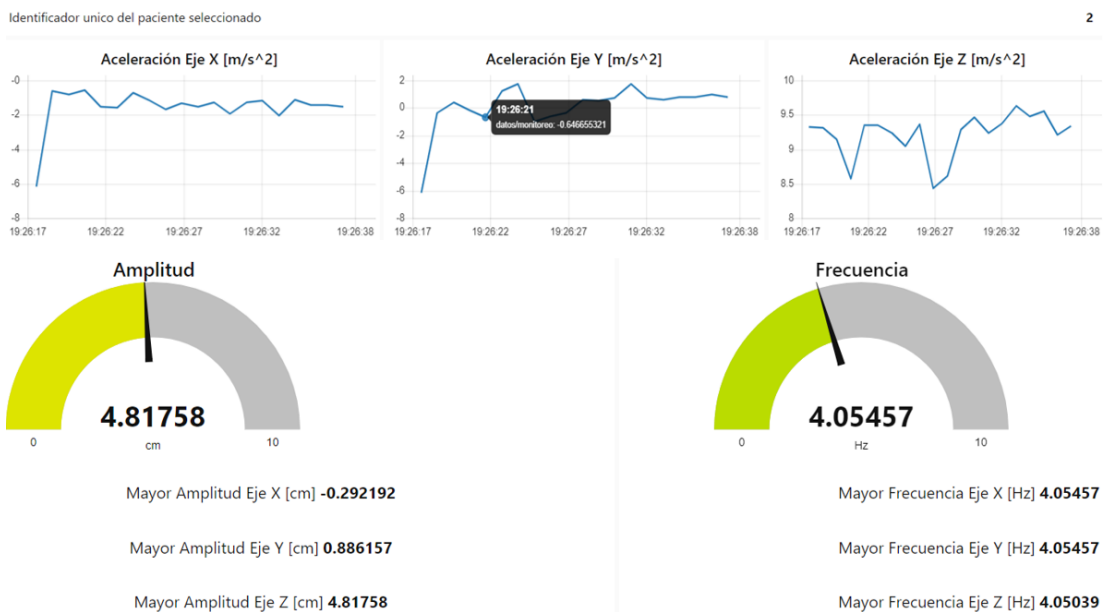
En la figura 72 se representan los resultados del TEST 2 realizado al paciente NN3 con EP.



**Figura 72. Resultados del TEST 2 del paciente NN3 con EP.**

**Elaborado por: investigador**

En la figura 73 se muestran los resultados del TEST 3 realizado al paciente 1 con EP.



**Figura 73. Resultados del TEST 3 del paciente NN3 con EP.**

**Elaborado por: investigador**



Los resultados de cada TEST del paciente NN3 con EP se presentan en la tabla 13.

**Tabla 13. Resultado de la evaluación realizada al paciente NN3 con EP**

<b>PACIENTE NN3 CON EP</b>								
<b>Identificador</b>	<b>AMPLITUD MAXIMA</b>				<b>FRECUENCIA MAXIMA</b>			
	Eje	TEST 1	TEST 2	TEST 3	Eje	TEST 1	TEST 2	TEST 3
3	X[cm]	0,89	2,15	-0,29	X [Hz]	0	4,01	4,05
	Y [cm]	0,80	2,48	0,88	Y [Hz]	4,03	4,07	4,05
	Z [cm]	5,92	5,20	4,81	Z [Hz]	4,05	4,04	4,05

**Elaborado por: investigador**

Al realizar el TEST 1 el paciente NN3 con EP presenta variaciones en las lecturas de amplitud de los ejes X y Y, debido a que sufre temblores en sus manos y sus valores son cercanos a 1.

En el TEST 2 el paciente NN3 con EP presenta variaciones en sus lecturas de amplitud de los ejes X y Y con valores de 2,15 y 2,48, debido a que al realizar estos movimientos de registran los temblores en sus manos por lo que los valores de aceleración aumentan y sus valores son mayores a los del paciente sin EP.

En el TEST 3 el paciente con NN3 EP presenta un incremento significativo con respecto al paciente sin EP en las lecturas de amplitud de los ejes X, Y, cuyas lecturas tienen valores de -2,902 y 0,886, debido a que la posición del brazo influye en las aceleraciones de los ejes y se registran los temblores en sus manos, por lo que los valores de aceleración aumentan y con ello los valores de amplitud.

En el eje Z, para los tres TESTs los valores están en un rango de 4,81 y 5,92 debido de dicho eje tiene una aceleración constante cercana a la gravedad de la tierra de  $9,8 \text{ m/s}^2$ .

En cuanto al valor de la frecuencia se obtiene que en los tres TESTs se encuentra en un rango de frecuencia de 0 a 4,07 Hz debido a que existió un incremento en el valor de las amplitudes, esto debido a que la frecuencia depende del valor de la amplitud calculada.

## Resultados de la evaluación realizada al paciente NN4 con EP

En la figura 74 se representan los resultados del TEST 1 realizado al paciente NN4 con EP.

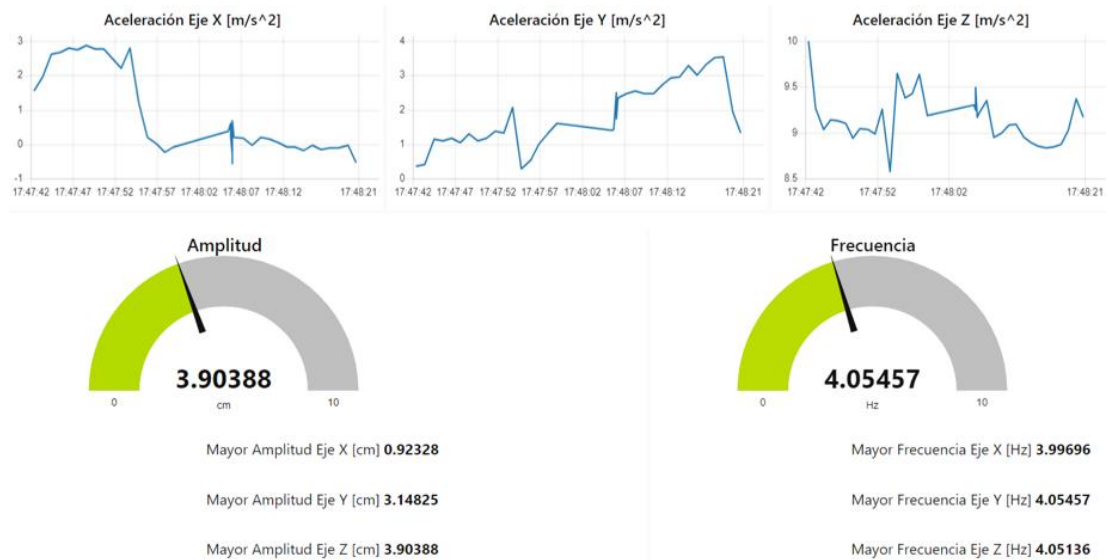


Figura 74. Resultados del TEST 1 del paciente NN4 con EP.

Elaborado por: investigador

En la figura 75 se representan los resultados del TEST 2 realizado al paciente NN4 con EP.

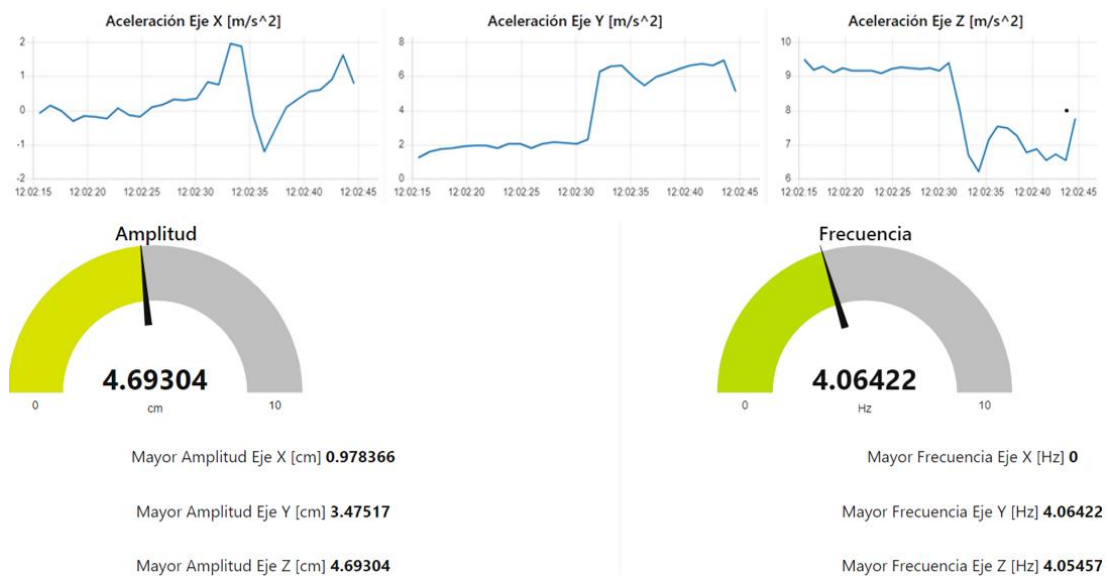
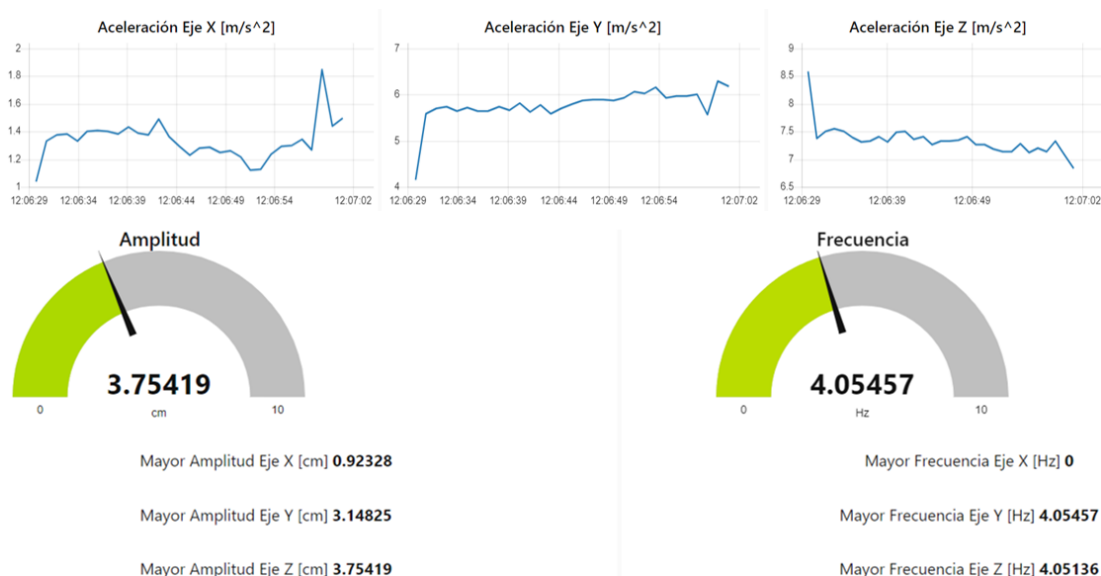


Figura 75. Resultados del TEST 2 del paciente NN4 con EP.

Elaborado por: investigador

En la Figura 76 se muestran los resultados del TEST 3 realizado al paciente NN4 con EP.



**Figura 76. Resultados del TEST 3 del paciente NN4 con EP.**

**Elaborado por: investigador**

Los resultados de cada TEST del paciente NN4 con EP se presentan en la tabla 14.

**Tabla 14. Resultados de la evaluación realizada al paciente NN4 con EP**

PACIENTE NN4 CON EP								
Identificador	AMPLITUD MAXIMA				FRECUENCIA MAXIMA			
	Eje	TEST 1	TEST 2	TEST 3	Eje	TEST 1	TEST 2	TEST 3
4	X [cm]	0,92	0,97	0,92	X [Hz]	3,99	0	0
	Y [cm]	3,14	3,47	3,14	Y [Hz]	4,05	4,06	4,05
	Z [cm]	3,90	4,69	3,75	Z [Hz]	4,05	4,05	4,05

**Elaborado por: investigador**

Durante el TEST 1, el paciente NN4 con EP presentó variaciones en las lecturas de amplitud de los ejes X y Y. Su rango de valores oscila entre 0,92 y 3,90. Debido a que el paciente tiene una edad avanzada, los temblores son más intensos, lo que aumenta el rango de la amplitud.

En el TEST 2, el paciente NN4 con EP presentó variaciones en las lecturas de amplitud de los ejes X y Y con valores que oscilaron entre 0,97 y 3,47. Estas variaciones se debieron a que los temblores en sus manos afectaron los movimientos realizados, lo que aumentó los valores de aceleración y los hizo mayores en comparación con los valores de un paciente sin EP.

Durante el TEST 3, el paciente NN4 con EP presentó un aumento significativo en comparación con un paciente sin EP en las lecturas de amplitud de los ejes X y Y, con valores que variaron entre 0,92 y 3,14. La posición del brazo influyó en las aceleraciones de los ejes, y los temblores en sus manos también aumentaron los valores de aceleración, lo que se reflejó en un aumento de los valores de amplitud

En el eje Z, durante los tres TESTs, los valores se encuentran en un rango de 3,90 a 4,69. Esto se debe a que dicho eje experimenta una aceleración constante cercana a la gravedad de la tierra, que es de  $9,8 \text{ m/s}^2$ .

En cuanto al valor de la frecuencia, se observó que durante los tres TESTs se encontraba en un rango de 0 a 4,05 Hz. Este aumento se debió a que también hubo un incremento en los valores de las amplitudes, ya que la frecuencia está relacionada con el valor de la amplitud calculada.

Cabe destacar que las pruebas se llevaron a cabo en ambas extremidades superiores, pero dado que los valores eran similares, se elaboró una tabla en general para cada prueba.

### **3.10.6 Resultado del ejercicio de terapia ocupacional**

En esta sección se presentan los resultados del ejercicio de terapia convencional, que consiste en trasladar esferas de un lugar a otro utilizando una pinza. Las esferas se colocarán a una distancia de 40 cm de separación, como se puede observar en la figura 77.



**Figura 77. Ejercicio de terapia ocupacional**

**Elaborado por: investigador**

Los dos participantes con EP tuvieron 1 sesión diaria durante 10 días con el sistema de rehabilitación propuesto y al finalizar las pruebas se comparó el tiempo de la primera sesión con la última sesión. A sí mismo, se realizó 5 sesiones a los pacientes sanos para tener un tiempo de referencia respecto a los pacientes con EP.

#### ***3.1.4.1 Resultados del ejercicio de terapia ocupacional realizado por los pacientes sanos y con EP***

En la tabla 15 se muestra los resultados de las pruebas medidos en minutos, segundos y milisegundos empleado por los pacientes sanos NN1 y NN2.

**Tabla 15. Tiempo empleado para realizar la terapia ocupacional por los pacientes sanos NN1 y NN2**

<b>TIEMPO EMPLEADO POR DE PACIENTES SANOS PARA REALIZAR LA TERAPIA OCUPACIONAL</b>												
<b>Pacientes</b>	<b>Paciente Sano NN1</b>						<b>Paciente Sano NN2</b>					
<b>Mano</b>	<b>Derecha</b>			<b>Izquierda</b>			<b>Derecha</b>			<b>Izquierda</b>		
	<b>mi</b>	<b>s</b>	<b>ms</b>	<b>mi</b>	<b>s</b>	<b>ms</b>	<b>mi</b>	<b>s</b>	<b>ms</b>	<b>mi</b>	<b>s</b>	<b>ms</b>
	<b>n</b>			<b>n</b>			<b>n</b>			<b>n</b>		
<b>Dia 1</b>	0	4	875	0	3	525	0	4	111	0	3	455
<b>Dia 2</b>	0	4	883	0	2	458	0	3	673	0	4	415
<b>Dia 3</b>	0	3	774	0	3	775	0	4	245	0	2	553
<b>Dia 4</b>	0	2	571	0	4	871	0	4	476	0	2	797
<b>Dia 5</b>	0	2	871	0	3	555	0	3	981	0	3	855
<b>Promedio</b>	0	3	794,8	0	3	636,8	0	3,6	497,2	0	2,8	615

**Elaborado por: investigador**

Para los pacientes sanos solo se realizó una sesión diaria durante 5 días, debido a que las mediciones no variaron. Como se era de esperar los pacientes sanos NN1 y NN2 no encontraron dificultad en llevar a cabo la actividad y lograron completarla en un tiempo récord con una media de 3 segundos con 600 milisegundos, demostrando la eficiencia y habilidad de los pacientes.

En la tabla 16 se muestra el tiempo empleado por los pacientes (NN3 y NN4 con EP) en cada una de las sesiones.

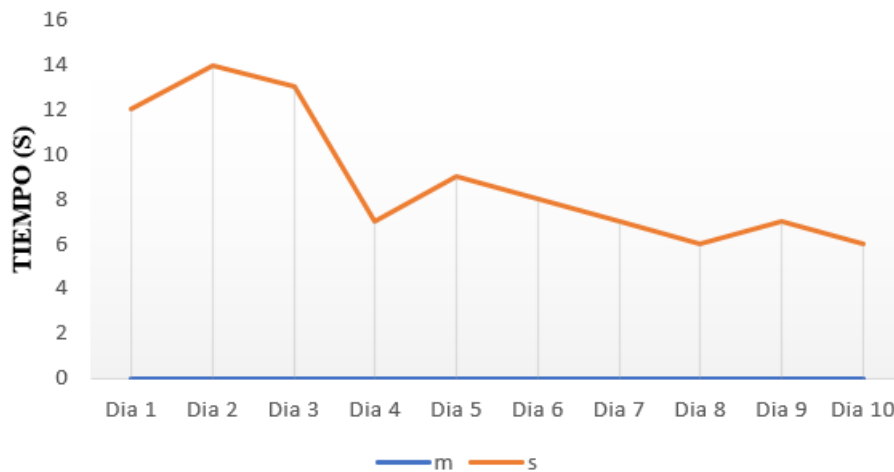
**Tabla 16. Tiempo empleado para realizar la terapia ocupacional por los pacientes NN3 y NN4 con EP**

<b>TIEMPO EMPLEADO POR PACIENTES CON EP PARA REALIZAR LA TERAPIA OCUPACIONAL</b>												
<b>Pacientes</b>	<b>Paciente NN1 con EP</b>						<b>Paciente NN2 con EP</b>					
<b>Mano</b>	<b>Mano Derecha</b>			<b>Mano Izquierda</b>			<b>Mano Derecha</b>			<b>Mano Izquierda</b>		
	<b>mi</b>	<b>s</b>	<b>ms</b>	<b>mi</b>	<b>s</b>	<b>ms</b>	<b>mi</b>	<b>s</b>	<b>ms</b>	<b>mi</b>	<b>s</b>	<b>ms</b>
Dia 1	0	12	454	0	13	454	0	20	995	0	22	566
Dia 2	0	14	788	0	13	875	0	21	578	0	22	665
Dia 3	0	13	522	0	12	774	0	19	335	0	18	556
Dia 4	0	7	777	0	8	412	0	14	885	0	8	558
Dia 5	0	9	532	0	8	797	0	13	665	0	8	856
Dia 6	0	8	625	0	7	412	0	13	561	0	8	456
Dia 7	0	7	652	0	6	998	0	10	225	0	7	556
Dia 8	0	6	321	0	6	553	0	8	885	0	6	526
Dia 9	0	7	988	0	6	885	0	8	556	0	6	875
Dia 10	0	6	741	0	6	755	0	8	445	0	6	556

**Elaborado por: investigador**

Es importante mencionar que la terapia se realizó en las extremidades superiores por separado, mano izquierda y mano derecha.

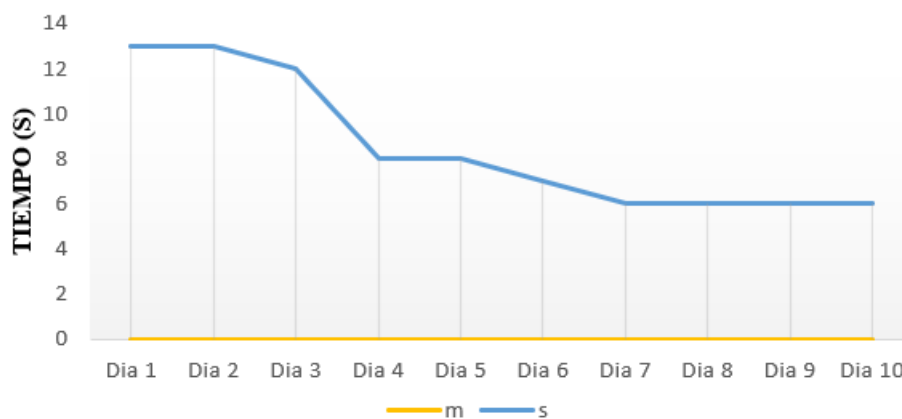
A continuación, se presentan los resultados obtenidos del paciente NN3 con EP; como se puede observar en la figura 78, en la primera sesión con la mano derecha se obtuvo un tiempo de 12 segundos, lo que está fuera del rango promedio en comparación con los pacientes sanos. Por lo que se realizó más sesiones hasta encontrar mejoras; entonces, en la cuarta sesión se obtuvo un tiempo de 7 segundos. Por último, en la sesión 10 logró un tiempo de 6 segundos, dando a notar que el paciente ha evolucionado significativamente en sus terapias. En la figura 78 se puede observar la mejoría de los tiempos durante las sesiones de terapia realizado.



**Figura 78. Gráfica evolutiva en la terapia convencional de motricidad fina de la mano derecha del paciente NN3**

**Elaborado por: investigador**

Los resultados del paciente NN3 con EP se presentan a continuación. En la primera sesión con la mano izquierda, obtuvo un tiempo de 13 segundos, que se encuentra por fuera del rango promedio en comparación con pacientes sanos, por ello se continuaron con más sesiones hasta observar mejorías. En la cuarta sesión, el paciente logró un tiempo de 8 segundos, y en la décima sesión logró un tiempo de 6 segundos, demostrando una evolución significativa en sus terapias. La figura 79 ilustra la mejoría en los tiempos durante las sesiones de terapia realizadas.



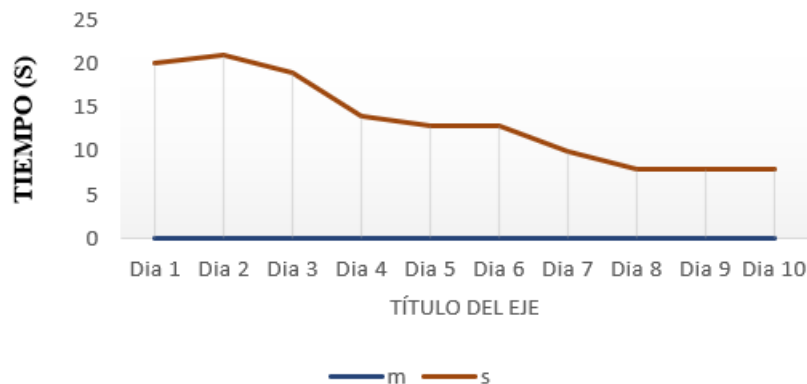
**Figura 79. Gráfica evolutiva en la terapia convencional de motricidad fina de la mano izquierda del paciente NN3**

**Elaborado por: investigador**

De igual forma, se expone la figura 80 donde se puede visualizar los avances del paciente NN4 con EP, pues en su primera sesión con la mano derecha logro realizar la



actividad en un tiempo de 20 segundos, el cual es alto debido y esto se debió a que la paciente tiene una edad y enfermedad avanzada. De igual forma se realizó más sesiones, mostrando mejorías en la sesión 7, pues consiguió un tiempo de 10 segundos, y en la última sesión alcanzó un tiempo de 8 segundos.



**Figura 80.** Gráfica evolutiva en la terapia convencional de motricidad fina de la mano derecha del paciente NN4

Elaborado por: investigador

Se puede observar en la figura 81 de los progresos del paciente NN4 con EP. En la primera sesión con la mano izquierda logró realizar la actividad en un tiempo de 22 segundos. Se realizó más sesiones, mostrando mejorías en la sesión 4 que consiguió un realizar en un tiempo de 14 segundos. En la última sesión logró realizar la actividad en un tiempo de 6 segundos.



**Figura 81.** Gráfica evolutiva en la terapia convencional de motricidad fina de la mano izquierda del paciente NN4

Elaborado por: investigador

En resumen, los resultados obtenidos en las pruebas de pacientes sanos y pacientes con Enfermedad de Parkinson muestran diferencias en cuanto a la amplitud y frecuencia de los temblores de las extremidades. Los pacientes sanos presentaron valores de amplitud en un rango de 0 a 1,5 cm y frecuencia de 4Hz, mientras que los pacientes con EP presentaron un aumento en la amplitud con un rango de valores de 0,80 a 5,92 y frecuencia de 4,07 Hz, lo que coincide con los rangos de frecuencia de los tipos de temblores presentados en la tabla 1. En los tres TESTs, los pacientes con EP presentaron un incremento significativo en las lecturas de aceleración en los tres ejes, lo cual produjo un aumento en la amplitud y frecuencia, demostrando que existe una relación de la aceleración con la frecuencia y amplitud de los temblores. Estos resultados son coherentes con la patología de la Enfermedad de Parkinson y son útiles en el diagnóstico y tratamiento de esta enfermedad.

Los resultados obtenidos de las pruebas del sistema de rehabilitación demuestran la eficacia de la terapia convencional para mejorar el manejo de la motricidad fina y el fortalecimiento de los músculos en pacientes con Parkinson. Se realizaron pruebas en pacientes sanos y con Parkinson, y se encontró que los pacientes sanos lograron completar la actividad en un tiempo de 3,1 segundos, mientras que los pacientes con Parkinson tardaron un tiempo promedio de 16,75 segundos en realizar la actividad en la primera sesión. Sin embargo, a medida que se continuó con la terapia, se observaron mejoras significativas en los tiempos para completar la actividad por parte de los pacientes con Parkinson. En la última sesión alcanzaron a completar la actividad en un tiempo promedio de 6,5 segundos, lo que demuestra que este dispositivo de terapia convencional es una herramienta eficaz para mejorar la movilidad en pacientes con Parkinson, y que las mejoras se producen a medida que se realizan más sesiones de terapia.

### **3.12 Presupuesto**

Para el cálculo del presupuesto en la implementación del sistema electrónico de rehabilitación y monitoreo para personas que padecen la enfermedad de Parkinson se tomó en cuenta el presupuesto de construcción y el de diseño. Para obtener el presupuesto de diseño se consideró el número de horas empleadas en su diseño y construcción y el salario de un Ingeniero en Telecomunicaciones que de acuerdo con

el Ministerio de Trabajo es de 858 dólares. Considerando un promedio de 21 días laborables, y utilizando la siguiente ecuación se calculó el salario por día [37].

$$Salario_{diario} = \frac{Salario_{mensual}}{Días_{laborables}}$$

$$Salario_{diario} = \frac{858}{21}$$

$$Salario_{diario} = \$40.86 [37]$$

El Ministerio de trabajo establece que por día son 8 horas laborables y aplicados a la siguiente ecuación se obtiene el valor en dólares por hora de trabajo.

$$Salario_{hora} = \frac{Salario_{diario}}{Horas_{laborables}}$$

$$Salario_{hora} = \frac{40.86}{8}$$

$$Salario_{hora} = \$5.11 [37]$$

Se estiman unas 320 horas de investigación utilizadas en el diseño, simulación, construcción y validación del funcionamiento: mediante la siguiente ecuación se obtuvo el presupuesto de diseño.

$$Presupuesto_{diseño} = Horas_{investigación} * Salario_{hora}$$

$$Presupuesto_{diseño} = 320 * 5.11$$

$$Presupuesto_{diseño} = \$ 1635,20$$

Los materiales utilizados en el desarrollo del proyecto se detallan en la tabla 17.

**Tabla 17. Presupuestos del proyecto**

N°	Detalle	Cantidad	Valor Unitario (USD)	Valor Total (USD)
1	Módulo ESP8266	2	14,00	28,00
2	TP4056	2	2,5	5,00
3	MPU6050	1	8,00	8,00
4	LCD 16x2	1	9,00	9,00
5	Batería Litio 3.7V	2	10,00	20,00
6	Case (Horas de diseño)	16	5,00	80,00
7	Case (Impresión)	3	8,00	24,00
8	Impresión PCB	2	20,00	40,00
9	Botones	4	0,50	2,00
10	Switch	3	0,60	1,80
11	Materiales de oficina	1	10,00	10,00
12	Servicio en la nube	-	12,26	12,26
13	Cable USB	1	3,50	3,50
14	Soporte de mano para dispositivo	1	10,00	10,00
15	Transporte	-	30,00	30,00
16	Otros gastos imprevistos	-	80,00	80,00
			Subtotal	363,56
			IVA (12%)	43,6272
			TOTAL	407,19

**Elaborado por: el investigador**

Con el uso de la siguiente ecuación se obtuvo el valor del presupuesto total.

$$Presupuesto_{total} = Presupuesto_{diseño} + Presupuesto_{construcción}$$

$$Presupuesto_{total} = \$1635,20 + 407,19\$$$

$$Presupuesto_{total} = \$ 2042,39$$

El presupuesto del proyecto fue de \$ 2042,39 y fue financiado en su totalidad por el tesista.

## CAPITULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1. Conclusiones

- Se determinó que la EP es un trastorno neurodegenerativo y afecta el sistema nervioso de muchas personas en todo el mundo. Los síntomas de esta enfermedad incluyen temblores, rigidez muscular, lentitud de movimiento, dificultad para mantener el equilibrio y cambios de personalidad. La comprensión de la característica y los síntomas de la enfermedad de Parkinson es fundamental para proporcionar un tratamiento adecuado a los pacientes que le permitan llevar una buena calidad de vida.
- La combinación de medicamentos y terapias de rehabilitación es el tratamiento más común para el manejo de la EP. En este proyecto se revisaron diferentes terapias, siendo las más destacadas la terapia ocupacional, la fisioterapia y la logopedia. La terapia ocupacional está indicada para mejorar la autonomía en autocuidado mediante actividades de la vida diaria y juegos adaptados para mejorar el equilibrio y la coordinación de sus movimientos, mientras que la fisioterapia es controlada diariamente por el fisioterapeuta a cargo y ayuda a mejorar la función motora de las extremidades y músculos. Por otro lado, la logopedia busca mejorar el habla y la pronunciación mediante sistemas de voz digitalizadas, mensajes de voz grabados o dispositivos de retroalimentación por ordenador portátil.
- Se desarrolló un dispositivo electrónico utilizando software de código abierto y elementos electrónicos disponibles en el mercado y de fácil adquisición. El sistema consta de un módulo destinado al monitoreo y está constituido de una tarjeta ESP8266 y un sensor MPU6050 que permite medir la aceleración de los movimientos de las extremidades superiores, con lo cual se puede determinar la amplitud y frecuencia de dichos movimientos, valores que le permiten al terapeuta establecer el estadio de la enfermedad y sus opciones de tratamiento; la información del monitoreo se almacena en un sistema gestor de base de datos MariaDB, los mismos que son presentados en diferentes interfaces gráficas. Para el proceso de rehabilitación se desarrolló un dispositivo electrónico, el

mismo que consta de un microcontrolador NodeMCU e interruptores de posición que juegan un papel fundamental en el control de la ubicación y el tiempo de terapia de fortalecimiento muscular y manejo de la motricidad fina.

- Una vez realizado las pruebas de funcionamiento del “Sistema electrónico de rehabilitación y monitoreo para personas que padecen la Enfermedad de Parkinson” se determinó que el dispositivo es fiable para el monitoreo y seguimiento del paciente debido a que proporciona una frecuencia de 4 Hz y valores de amplitud mayores a 1 en los temblores de las extremidades superiores de los pacientes con EP. Además, el dispositivo electrónico aporta significativamente a los procesos de rehabilitación ya que se observó una mejoría en el tiempo para realizar los ejercicios, pasando de 16.75 segundos a 6.5 segundos, a medida que se avanzaba con las sesiones diarias realizadas a los pacientes con EP.

#### **4.2. Recomendaciones**

- Para futuras mejoras del prototipo se recomienda cambiar el microcontrolador por un microordenador, ya que estos tienen más funciones como la de utilizar otros lenguajes de programación que sean más específicos para procesar esta información, pues cuentan con la posibilidad de integrar en ellos mismo softwares de código abierto sin la necesidad de utilizar una plataforma de cloud computing, lo que permite ahorrar costos de implementación y una mayor velocidad de procesamiento de los datos.
- Se recomienda implementar más dispositivos electrónicos de terapia que ayuden a realizar mayor variedad de ejercicios de terapia ocupacional. De esta manera se podrá combinar los ejercicios durante las sesiones de terapia y así obtener una mejor evolución del paciente en su rehabilitación.
- Para ahorrar el consumo de energía, mejorar la presentación de resultados en pantalla y reducir el tamaño del prototipo de rehabilitación se recomienda utilizar una pantalla OLED, ya que esta presenta mayor eficiencia energética, brillo y nitidez de los colores en comparación a la LCD 16x2 que se utilizó en el presente proyecto.

## BIBLIOGRAFIA

- [1] Organización Mundial de la Salud, Trastornos neurológicos: desafíos para la salud pública, Ginebra: World Health Organization, 2006.
- [2] Aesthesis Psicólogos, "psicologos madrid capital," 11 04 2022. [Online]. Available: <https://www.psicologosmadridcapital.com/blog/parkinson-y-sus-consecuencias/>. [Accessed 21 07 2022].
- [3] "Fundacion Degen," 08 04 2021. [Online]. Available: <https://conoceelparkinson.org/cuidados/dia-mundial-parkinson-2021/>. [Accessed 2022 06 06].
- [4] M. Aliño Costa, "Universidad Internacional de Valencia," 20 08 2020. [Online]. Available: <https://www.universidadviu.com/ec/actualidad/nuestros-expertos/parkinson-y-neuropsicologia#:~:text=Los%20pa%C3%ADses%20con%20un%20aumento,son%20Venezuela%2C%20Cuba%20y%20Uruguay..>
- [5] J. P. Montalvo Herdoíza, P. S. Montalvo Perero, . L. E. Albear Toala, E. R. Intriago Mercado and D. V. Moreira Vera, "Prevalencia de la Enfermedad de Parkinson: Estudio Puerta-Puerta en la Provincia de Manabí-Ecuador," *Rev. Ecuat. Neurol*, p. 4, 2017.
- [6] Honorable Gobierno Provincial de Tungurahua , 25 06 2018. [Online]. Available: <https://www.tungurahua.gob.ec/index.php/informativo-hgpt/principales/3227-en-la-campana-de-prevencion-de-parkinson-y-alzheimer>. [Accessed 30 05 2022].
- [7] R. García Ramos , "Asociación Parkinson Madrid," [Online]. Available: <https://www.parkinsonmadrid.org/el-parkinson/el-parkinson-tratamiento/>.
- [8] F. J. Chacha Cajamarca, "Diseño, desarrollo e implementación de un dispositivo inalámbrico para estimular ejercicios de rehabilitación en pacientes con Parkinson," Universidad Politécnica Salesiana sede Cuenca, Cuenca, 2017.
- [9] B. C. Punin Sigcha , "Diseño, desarrollo e implementación de un dispositivo inalámbrico para detección de episodios de congelamiento de la marcha en

pacientes con la enfermedad de Parkinson," Universidad Politecnica Salesiana , Cuenca, 2017.

- [10] F. A. Fajardo Villanueva, "Dispositivo electrónico para la adquisición de señales asociadas a la enfermedad de Parkinson (EP)," Departamento de Ciencias de la Computación y Electrónica , Barranquilla, 2018.
- [11] V. G. Lechon Anzules , "Dispositivo para medir el tremor en personas con la enfermedad de Parkinson," Universidad Tecnica del norte , Ibarra, 2019.
- [12] G. Trinidad Barnech, "Dispositivo de asistencia ante el congelamiento de la marcha en pacientes con Parkinson," Universidad de la Republica, Montevideo , 2020.
- [13] L. F. Sigcha Guachamin , "Detección automática de síntomas motores asociados a la Enfermedad de Parkinson mediante relojes y teléfonos móviles inteligentes aplicando técnicas de inteligencia artificial," Universidad Politecnica de Madrid, Madrid, 2021.
- [14] C. R. Godinez Tello and L. E. Pro Concepcion , "Una aplicación móvil con técnicas de machine learning a partir de datos de sensores inerciales para el monitoreo e identificación de ocurrencias del temblor de Parkinson," Universidad Nacional Mayor de San Marcos, Lima, 2021.
- [15] . P. Werner, S. Klaus, C. Tanner, G. Halliday and P. Brundin, "Parkinson disease," *Nat Rev Dis Primers*, vol. 3, no. 17013, p. 21, 2017.
- [16] B. M. Chacon and K. F. Robles, "Diseño e implementación de un dispositivo de estimulación vibratoria para pacientes con Parkinson y problemas de congelamiento de marcha," Universidad del Azuay , Cuenca, 2016.
- [17] M. clinic, "Enfermedad de Parkinson," Bayo Clinic, Estados unidos , 2021.
- [18] ParkinsonbahiaCadiz.Org, "Parkinson Bahía de Cádiz," 30 07 2019. [Online]. Available: <https://parkinsonbahiaCadiz.org/taller-de-terapia-ocupacional/>. [Accessed 01 11 2022].



- [19] V. Marquez , "Asociación Parkinson Elche," 22 07 2021. [Online]. Available: <https://parkinsonelche.es/ejercicio-de-motricidad-fina/>. [Accessed 05 11 2022].
- [20] W. Quispe, "Aprendiendo con Arduino," 11 09 2018. [Online]. Available: <https://aprendiendoarduino.wordpress.com/2016/11/06/electronica-sensores-actuadores-y-perifericos/>. [Accessed 2022 05 30].
- [21] Sensing, "Sensing," [Online]. Available: <https://sensores-de-medida.es/medicion/sensores-y-transductores/acelerometros-sensores-de-aceleracion/acelerometros-triaxiales/>. [Accessed 22 07 2022].
- [22] naylampmechatronics., "Tutorial MPU6050, acelerómetro y giroscopio," naylampmechatronics., Lima.
- [23] E. p. focus, "Electronics projects focus," [Online]. Available: <https://www.elprocus.com/gyroscope-sensor/>. [Accessed 2022 07 07].
- [24] Azure, "Azure," [Online]. Available: <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-cloud-storage/>. [Accessed 07 07 2022].
- [25] L. E. Villarroel Morales, "Sistema de estimulación eléctrica para los pacientes que padecen de pie equino del Centro de Rehabilitación Física y Neurológica Bendiciones," Universidad Técnica de Ambato, Ambato, 2021.
- [26] "Unit Electronics," [Online]. Available: <https://uelectronics.com/producto/bateria-lipo-3-7v-1500mah-103050/>. [Accessed 02 01 2023].
- [27] "Electro Store," [Online]. Available: <https://grupoelectrostore.com/shop/modulos-y-shields/cargadores-para-baterias/modulo-tp4056-5v-micro-usb-1a-cargador-para-baterias-litio-con-proteccion/>. [Accessed 20 01 2023].
- [28] J. L. Castañeda Sandoval, "Sistema de Detección y Monitoreo Transcutáneo no invasivo de Hiperbilirrubinemia en Neonatos," Universidad Técnica de Ambato, Ambato, 2022.

- [29] A. L. Alarcón Ortiz, "Sistema de monitorización y control de ganado vacuno por medio de tecnología inalámbrica para prevención de abigeato," Universidad Técnica de Ambato, Ambato, 2018.
- [30] P. Wankhede, T. Minaiy and R. Chinchamatpure, "COMPARATIVE STUDY OF CLOUD PLATFORMS MICROSOFT AZURE, GOOGLE CLOUD PLATFORM AND AMAZON EC2," UG students, Department of Computer Engineering , India, 2020.
- [31] E. J. Guananga Narvaez and W. R. VIVAS DIAZ , "COMPARACIÓN DE PROTOCOLOS DE COMUNICACIÓN PARA INTERNET DE AS COSAS (IOT)," UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO, Quito, 2022.
- [32] "desarrolloweb.com," 19 07 2002. [Online]. Available: <https://desarrolloweb.com/articulos/840.php>. [Accessed 10 01 2023].
- [33] S. Romerson, "slideshare," 04 07 2018. [Online]. Available: <https://es.slideshare.net/RomersonSantana/oracle-database-100548612>. [Accessed 10 01 2023].
- [34] A. V. Bermeo Maldonado and M. F. Bravo Guamán, "Diseño y desarrollo de un sistema inalámbrico que permita monitorear los temblores en pacientes que padecen la enfermedad de Parkinson utilizando software y hardware libre," Universidad Politécnica Saleciana, Cuenca, 2016.
- [35] W. Tomasi, Sistemas de Comunicaciones Electrónicas, Mexico: Pearson Educación, 2003.
- [36] M. Sicre Marquez , "Estudio en una muestra de enfermos de Parkinson con temblor," Universidad de Jaén, Jaén, 2019.
- [37] D. C. Jiménez Zuñiga, "Sistema de ubicación y monitoreo de señales vitales en adultos mayores y personas con alzheimer," Universidad Técnica de Ambato, Ambato, 2018.

# **ANEXOS**

## Anexo 1. Validación Lcda. Liliana Gabriela Sunta Moreno

### CERTIFICACIÓN

Ingeniera, Mg.

Pilar Urrutia

DECANA

Facultad de Ingeniería en Sistemas, Electrónica e Industrial

Presente. –

Yo, Liliana Gabriela Sunta Moreno, con cédula de ciudadanía 1804619861 con registro del Senecyt N° 1010-15-1368793, Licenciada en Terapia Física, certifico que he revisado el proyecto técnico “SISTEMA ELECTRÓNICO DE REHABILITACIÓN Y MONITOREO PARA PERSONAS QUE PADECEN LA ENFERMEDAD DE PARKINSON”; el mismo que fue desarrollado en su total autoría por el Sr. Luis Humberto Cuchiye Diaz, portador de la Cédula de Ciudadanía 1501056426, estudiante de la Carrera de Telecomunicaciones de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato. Es así que después de haber realizado las pruebas necesarias me permito verificar la funcionalidad del equipo y puedo validar que los resultados obtenidos hasta la fecha son aceptables y de gran ayuda en el área de monitoreo y rehabilitación de la motricidad fina de las extremidades superiores de pacientes que padecen la Enfermedad de Parkinson.

Saludos cordiales,

Atentamente,

**Lcda. Liliana Sunta**  
**FISIOTERAPEUTA**  
**MSP-2017-Z03-0018369**

Lcda. Liliana Gabriela Sunta Moreno

Fisioterapeuta

CI: 1804619861

REG: 1010-15-1368793

## Anexo 2. Datasheed Tarjeta ESP8266

# Internet of Things

## NodeMCU ESP8266 ESP-12E WiFi Development Board

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the DevKit. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.



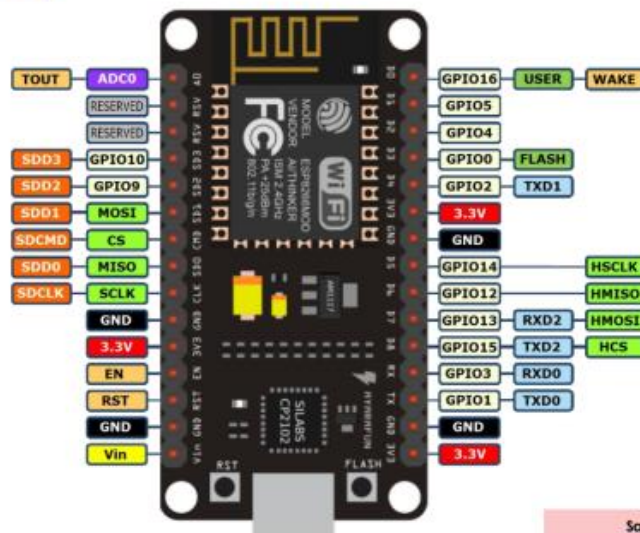
### Features

- ▶ Version : DevKit v1.0
- ▶ Breadboard Friendly
- ▶ Light Weight and small size.
- ▶ 3.3V operated, can be USB powered.
- ▶ Uses wireless protocol 802.11b/g/n.
- ▶ Built-in wireless connectivity capabilities.
- ▶ Built-in PCB antenna on the ESP-12E chip.
- ▶ Capable of PWM, I2C, SPI, UART, 1-wire, 1 analog pin.
- ▶ Uses CP2102 USB Serial Communication interface module.
- ▶ Arduino IDE compatible (extension board manager required).
- ▶ Supports Lua (alike node.js) and Arduino C programming language.



### PINOUT DIAGRAM

#### NodeMCU ESP8266 v1.0



Source  
<https://iotbytes.wordpress.com/nodemcu-pinout/>

**Safety Precaution**  
 All GPIO runs at 3.3V !!

## NodeMCU ESP8266



Front View



Front View

### Specifications of ESP-12E WiFi Module

<b>Wireless Standard</b>	IEEE 802.11 b/g/n
<b>Frequency Range</b>	2.412 - 2.484 GHz
<b>Power Transmission</b>	802.11b : +16 ± 2 dBm (at 11 Mbps) 802.11g : +14 ± 2 dBm (at 54 Mbps) 802.11n : +13 ± 2 dBm (at HT20, MCS7)
<b>Receiving Sensitivity</b>	802.11b : -93 dBm (at 11 Mbps, CCK) 802.11g : -85 dBm (at 54 Mbps, OFDM) 802.11n : -82 dBm (at HT20, MCS7)
<b>Wireless Form</b>	On-board PCB Antenna
<b>IO Capability</b>	UART, I2C, PWM, GPIO, 1 ADC
<b>Electrical Characteristic</b>	3.3 V Operated 15 mA output current per GPIO pin 12 - 200 mA working current Less than 200 uA standby current
<b>Operating Temperature</b>	-40 to +125 °C
<b>Serial Transmission</b>	110 - 921600 bps, TCP Client 5
<b>Wireless Network Type</b>	STA / AP / STA + AP
<b>Security Type</b>	WEP / WPA-PSK / WPA2-PSK
<b>Encryption Type</b>	WEP64 / WEP128 / TKIP / AES
<b>Firmware Upgrade</b>	Local Serial Port, OTA Remote Upgrade
<b>Network Protocol</b>	IPv4, TCP / UDP / FTP / HTTP
<b>User Configuration</b>	AT + Order Set, Web Android / IOS, Smart Link APP

## Anexo 3. Datasheed Sensor MPU6050

	<b>MPU-6000/MPU-6050 Product Specification</b>	Document Number: PS-MPU-6000A-00 Revision: 3.4 Release Date: 08/19/2013
---	--	---

### Primary Differences between MPU-6000 and MPU-6050

Part / Item	MPU-6000	MPU-6050
VDD	2.375V-3.46V	2.375V-3.46V
VLOGIC	n/a	1.71V to VDD
Serial Interfaces Supported	I <sup>2</sup> C, SPI	I <sup>2</sup> C
Pin 8	/CS	VLOGIC
Pin 9	AD0/SDO	AD0
Pin 23	SCL/SCLK	SCL
Pin 24	SDA/SDI	SDA

	<b>MPU-6000/MPU-6050 Product Specification</b>	Document Number: PS-MPU-6000A-00 Revision: 3.4 Release Date: 08/19/2013
--	--	---

## 5 Features

### 5.1 Gyroscope Features

The triple-axis MEMS gyroscope in the MPU-60X0 includes a wide range of features:

- Digital-output X-, Y-, and Z-Axis angular rate sensors (gyroscopes) with a user-programmable full-scale range of  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ , and  $\pm 2000^\circ/\text{sec}$
- External sync signal connected to the FSYNC pin supports image, video and GPS synchronization
- Integrated 16-bit ADCs enable simultaneous sampling of gyros
- Enhanced bias and sensitivity temperature stability reduces the need for user calibration
- Improved low-frequency noise performance
- Digitally-programmable low-pass filter
- Gyroscope operating current: 3.6mA
- Standby current: 5 $\mu$ A
- Factory calibrated sensitivity scale factor
- User self-test

### 5.2 Accelerometer Features

The triple-axis MEMS accelerometer in MPU-60X0 includes a wide range of features:

- Digital-output triple-axis accelerometer with a programmable full scale range of  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  and  $\pm 16g$
- Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer
- Accelerometer normal operating current: 500 $\mu$ A
- Low power accelerometer mode current: 10 $\mu$ A at 1.25Hz, 20 $\mu$ A at 5Hz, 60 $\mu$ A at 20Hz, 110 $\mu$ A at 40Hz
- Orientation detection and signaling
- Tap detection
- User-programmable interrupts
- High-G interrupt
- User self-test

## 6 Electrical Characteristics

### 6.1 Gyroscope Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
<b>GYROSCOPE SENSITIVITY</b> Full-Scale Range	FS_SEL=0		±250		°/s	
	FS_SEL=1		±500		°/s	
	FS_SEL=2		±1000		°/s	
	FS_SEL=3		±2000		°/s	
Gyroscope ADC Word Length Sensitivity Scale Factor	FS_SEL=0		16		bits	
	FS_SEL=1		131		LSB/(°/s)	
	FS_SEL=2		65.5		LSB/(°/s)	
	FS_SEL=3		32.8		LSB/(°/s)	
Sensitivity Scale Factor Tolerance	25°C	-3	16.4	+3	%	
	Sensitivity Scale Factor Variation Over Temperature		±2		%	
Nonlinearity	Best fit straight line; 25°C		0.2		%	
Cross-Axis Sensitivity			±2		%	
<b>GYROSCOPE ZERO-RATE OUTPUT (ZRO)</b>						
Initial ZRO Tolerance	25°C		±20		°/s	
ZRO Variation Over Temperature	-40°C to +85°C		±20		°/s	
Power-Supply Sensitivity (1-10Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		°/s	
Power-Supply Sensitivity (10 - 250Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		°/s	
Power-Supply Sensitivity (250Hz - 100kHz)	Sine wave, 100mVpp; VDD=2.5V		4		°/s	
Linear Acceleration Sensitivity	Static		0.1		°/s/g	
<b>SELF-TEST RESPONSE</b>						
Relative	Change from factory trim	-14		14	%	1
<b>GYROSCOPE NOISE PERFORMANCE</b>						
Total RMS Noise	FS_SEL=0 DLPCFG=2 (100Hz)		0.05		°/s-rms	
Low-frequency RMS noise	Bandwidth 1Hz to 10Hz		0.033		°/s-rms	
Rate Noise Spectral Density	At 10Hz		0.005		°/s/√Hz	
<b>GYROSCOPE MECHANICAL FREQUENCIES</b>						
X-Axis		30	33	36	kHz	
Y-Axis		27	30	33	kHz	
Z-Axis		24	27	30	kHz	
<b>LOW PASS FILTER RESPONSE</b>						
	Programmable Range	5		256	Hz	
<b>OUTPUT DATA RATE</b>						
	Programmable	4		8,000	Hz	
<b>GYROSCOPE START-UP TIME</b>						
ZRO Settling (from power-on)	DLPCFG=0 to ±1°/s of Final		30		ms	

1. Please refer to the following document for further information on Self-Test: *MPU-6000/MPU-6050 Register Map and Descriptions*

### 6.2 Accelerometer Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
<b>ACCELEROMETER SENSITIVITY</b> Full-Scale Range	AFS_SEL=0		±2		g	
	AFS_SEL=1		±4		g	
	AFS_SEL=2		±8		g	
	AFS_SEL=3		±16		g	
ADC Word Length	Output in two's complement format		16		bits	
Sensitivity Scale Factor	AFS_SEL=0		16,384		LSB/g	
	AFS_SEL=1		8,192		LSB/g	
	AFS_SEL=2		4,096		LSB/g	
AFS_SEL=3		2,048		LSB/g		
Initial Calibration Tolerance			±3		%	
Sensitivity Change vs. Temperature	AFS_SEL=0, -40°C to +85°C		±0.02		%/°C	
Nonlinearity	Best Fit Straight Line		0.5		%	
Cross-Axis Sensitivity			±2		%	
<b>ZERO-G OUTPUT</b>						
Initial Calibration Tolerance	X and Y axes Z axis		±50 ±80		mg mg	1
Zero-G Level Change vs. Temperature	X and Y axes, 0°C to +70°C Z axis, 0°C to +70°C		±35 ±60		mg	
<b>SELF TEST RESPONSE</b>						
Relative	Change from factory trim	-14		14	%	2
<b>NOISE PERFORMANCE</b>						
Power Spectral Density	@ 10Hz, AFS_SEL=0 & ODR=1KHz		400		μg/√Hz	
<b>LOW PASS FILTER RESPONSE</b>						
	Programmable Range	5		260	Hz	
<b>OUTPUT DATA RATE</b>						
	Programmable Range	4		1,000	Hz	
<b>INTELLIGENCE FUNCTION INCREMENT</b>						
			32		mg/LSB	

1. Typical zero-g initial calibration tolerance value after MSL3 preconditioning
2. Please refer to the following document for further information on Self-Test: *MPU-6000/MPU-6050 Register Map and Descriptions*



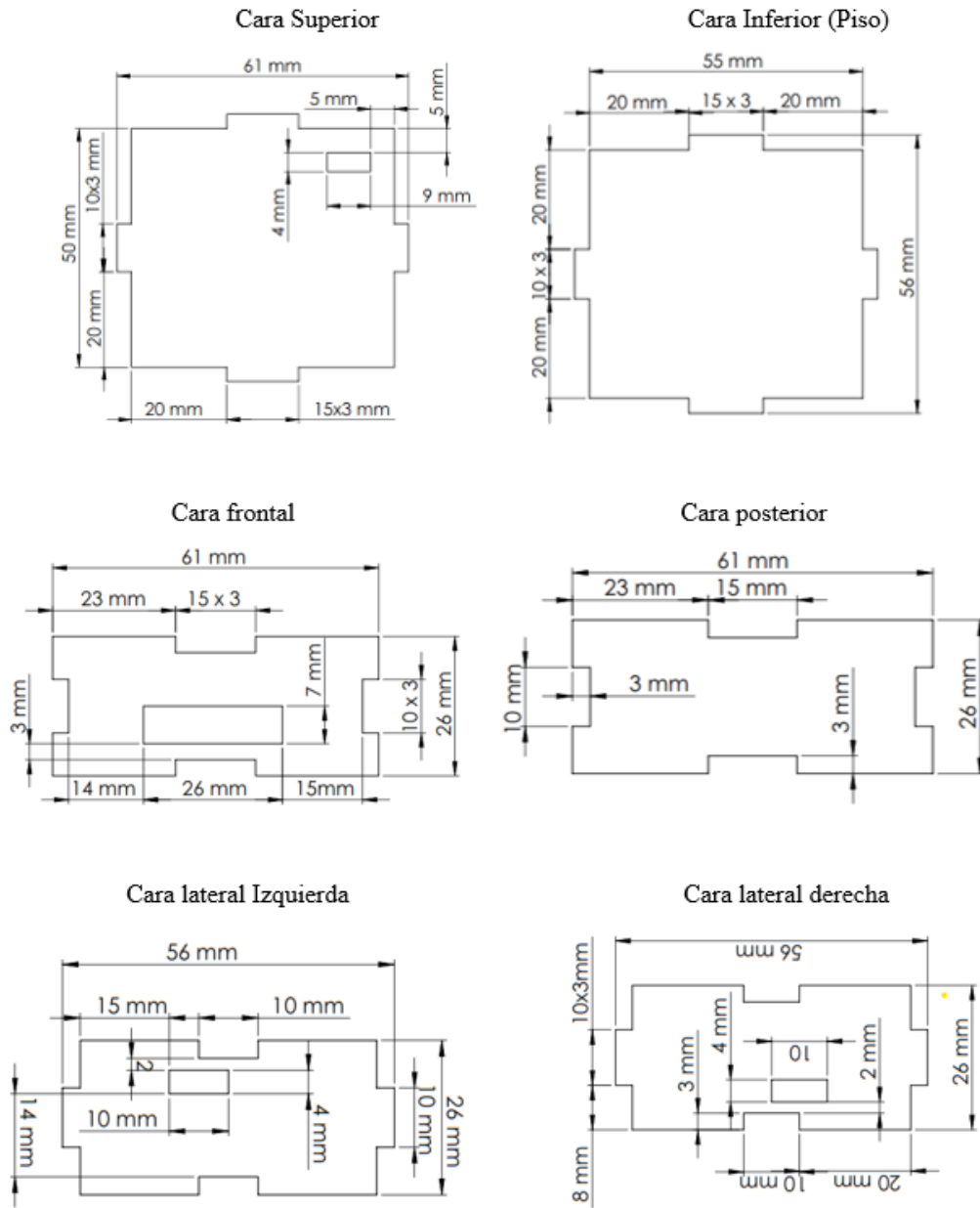


**6.3 Electrical and Other Common Specifications**

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C

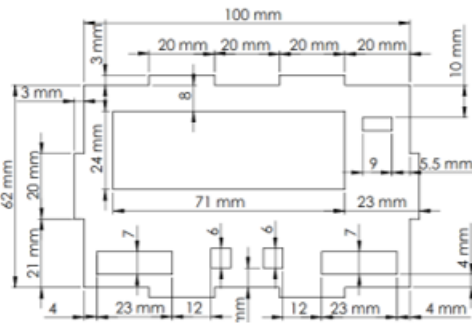
PARAMETER	CONDITIONS	MIN	TYP	MAX	Units	Notes
<b>TEMPERATURE SENSOR</b>						
Range			-40 to +85		°C	
Sensitivity	Untrimmed		340		LSB/°C	
Temperature Offset	35°C		-521		LSB	
Linearity	Best fit straight line (-40°C to +85°C)		±1		°C	
<b>VDD POWER SUPPLY</b>						
Operating Voltages		2.375		3.46	V	
Normal Operating Current	Gyroscope + Accelerometer + DMP		3.9		mA	
	Gyroscope + Accelerometer (DMP disabled)		3.8		mA	
	Gyroscope + DMP (Accelerometer disabled)		3.7		mA	
	Gyroscope only (DMP & Accelerometer disabled)		3.6		mA	
	Accelerometer only (DMP & Gyroscope disabled)		500		µA	
Accelerometer Low Power Mode Current	1.25 Hz update rate		10		µA	
	5 Hz update rate		20		µA	
	20 Hz update rate		70		µA	
	40 Hz update rate		140		µA	
Full-Chip Idle Mode Supply Current			5		µA	
Power Supply Ramp Rate	Monotonic ramp. Ramp rate is 10% to 90% of the final value			100	ms	
<b>VLOGIC REFERENCE VOLTAGE</b>						
Voltage Range	MPU-6050 only VLOGIC must be ≤VDD at all times	1.71		VDD	V	
Power Supply Ramp Rate	Monotonic ramp. Ramp rate is 10% to 90% of the final value			3	ms	
Normal Operating Current			100		µA	
<b>TEMPERATURE RANGE</b>						
Specified Temperature Range	Performance parameters are not applicable beyond Specified Temperature Range	-40		+85	°C	

## Anexo 4. Dimensiones de la carcasa del sistema electrónico de monitoreo

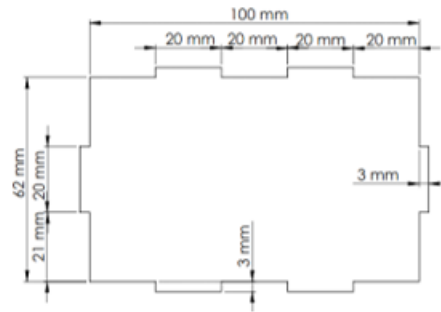


## Anexo 5. Dimensiones de la carcasa del sistema electrónico de rehabilitación

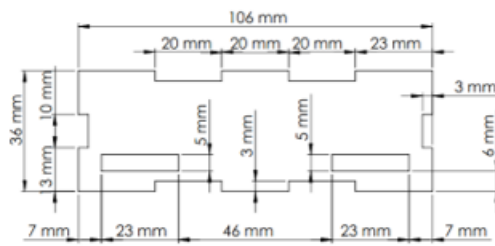
Cara Superior



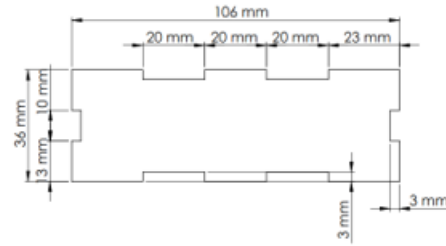
Cara Inferior (Piso)



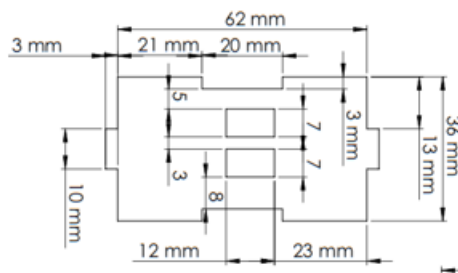
Cara frontal



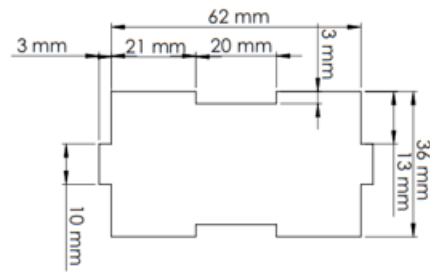
Cara posterior



Cara lateral Izquierda



Cara lateral derecha



## Anexo 6. Código de Arduino dispositivo de monitoreo

```
#include <WiFiManager.h>
#include <PubSubClient.h>
#include <Arduino.h>
#include <ArduinoJson.h>
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"
#include "arduinoFFT.h"

//Pines botones
#define button1 13 //D7
#define button2 12 //D6
#define button3 0 //D3
int count=0; //contador
#define ledPIN 14 //D5

WiFiClient espClient;
PubSubClient client(espClient);

//Dirección IP del broker MQTT para conectarse a este
const char* mqtt_server = "20.38.46.67";

arduinoFFT FFT = arduinoFFT(); /* Create FFT object */
const uint16_t samples = 64; //This value MUST ALWAYS be a power of 2
const double signalFrequency = 4;
const double samplingFrequency = 100;
//const uint8_t amplitude = 100;

double vReal[samples];
double vImag[samples];

//Variables MPU6050
```

```

const int mpuAddress = 0x68; // Puede ser 0x68 o 0x69
MPU6050 mpu(mpuAddress);
unsigned long lastMsg = 0;

//Variables Aceleración
int16_t ax, ay, az;
double acx = 0;
double acy = 0;
double acz = 0;

double tiempo = 3000; //ESTABLECER TIEMPO DE EJECUCION
//Variables Amplitud
double Ts = 0.1;
double sx = 0;
double sy = 0;
double sz = 0;

//Variables frecuencia
double fx = 0;
double fy = 0;
double fz = 0;

// Factores de conversion
const float accScale = 2.0 * 9.81 / 32768.0;

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Create a random client ID
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    // Attempt to connect

```

```

if (client.connect(clientId.c_str())) {
    Serial.println("connected");
    digitalWrite(ledPIN , HIGH);
    // Once connected, publish an announcement...
    client.publish("datos/estado", "Conectado");
    // ... and resubscribe
    client.subscribe("datos/#");
} else {
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    // Wait 5 seconds before retrying
    delay(5000);
}
}
}

void setup(){
    Serial.begin(115200);
    client.setServer(mqtt_server, 1883);
    WiFiManager wifiManager; // Creamos una instancia de la clase WiFiManager
    //wifiManager.resetSettings(); // Descomentar para resetear configuración
    wifiManager.autoConnect("ESP8266-LC"); // Creamos AP y portal cautivo
    Serial.println(";Conectado!");
    Wire.begin();
    mpu.initialize();
    Serial.println(mpu.testConnection() ? F("IMU iniciado correctamente") : F("Error al
iniciar IMU"));
    Serial.println(Ts);
    //iniciar botones en alto
    pinMode(button1,INPUT_PULLUP);
    pinMode(button2,INPUT_PULLUP);
    pinMode(button3,INPUT_PULLUP);

```

```

    pinMode(ledPIN , OUTPUT);
}
void aceleraciones_amplitudes(){
    // Leer las aceleraciones
    mpu.getAcceleration(&ax, &ay, &az);
    acx = ax * accScale;//random(1,10);
    acy = ay * accScale;//random(11,20);
    acz = az * accScale;//random(21,30);

    //Calcular amplitud
    sx = 50 * acx * (Ts*Ts);
    sy = 50 * acy * (Ts*Ts);
    sz = 50 * acz * (Ts*Ts);
}
void frecuenciax(){
    //Calculo FFT
    double amplitude = sx;
    /* Build raw data */
    double cycles = (((samples-1) * signalFrequency) / samplingFrequency); //Number
of signal cycles that the sampling will read
    for (uint16_t i = 0; i < samples; i++)
    {
        vReal[i] = int8_t((amplitude * (sin((i * (twoPi * cycles)) / samples))) / 2.0);/* Build
data with positive and negative values*/
        vImag[i] = 0.0;
    }
    /* Print the results of the simulated sampling according to time */
    FFT.Windowing(vReal,          samples,          FFT_WIN_TYP_HAMMING,
FFT_FORWARD); /* Weigh data */
    FFT.Compute(vReal, vImag, samples, FFT_FORWARD); /* Compute FFT */
    FFT.ComplexToMagnitude(vReal, vImag, samples); /* Compute magnitudes */
    fx = FFT.MajorPeak(vReal, samples, samplingFrequency);
    if(isnan(fx)){

```

```

        fx=0;}
    }
void frecuenciay(){
    //Calculo FFT
    double amplitude1 = sy;
    /* Build raw data */
    double cycles = (((samples-1) * signalFrequency) / samplingFrequency); //Number
of signal cycles that the sampling will read
    for (uint16_t i = 0; i < samples; i++)
    {
        vReal[i] = int8_t((amplitude1 * (sin((i * (twoPi * cycles)) / samples))) / 2.0);/*
Build data with positive and negative values*/
        vImag[i] = 0.0;
    }
    /* Print the results of the simulated sampling according to time */
    FFT.Windowing(vReal,          samples,          FFT_WIN_TYP_HAMMING,
FFT_FORWARD); /* Weigh data */
    FFT.Compute(vReal, vImag, samples, FFT_FORWARD); /* Compute FFT */
    FFT.ComplexToMagnitude(vReal, vImag, samples); /* Compute magnitudes */
    fy = FFT.MajorPeak(vReal, samples, samplingFrequency);
    if(isnan(fy)){
        fy=0;}
    }
void frecuenciaz(){
    //Calculo FFT
    double amplitude2 = sz;
    /* Build raw data */
    double cycles = (((samples-1) * signalFrequency) / samplingFrequency); //Number
of signal cycles that the sampling will read
    for (uint16_t i = 0; i < samples; i++)
    {
        vReal[i] = int8_t((amplitude2 * (sin((i * (twoPi * cycles)) / samples))) / 2.0);/*
Build data with positive and negative values*/

```



```

    vImag[i] = 0.0;
}
/* Print the results of the simulated sampling according to time */
FFT.Windowing(vReal,          samples,          FFT_WIN_TYP_HAMMING,
FFT_FORWARD); /* Weigh data */
FFT.Compute(vReal, vImag, samples, FFT_FORWARD); /* Compute FFT */
FFT.ComplexToMagnitude(vReal, vImag, samples); /* Compute magnitudes */
fz = FFT.MajorPeak(vReal, samples, samplingFrequency);
if(isnan(fz)){
    fz=0;}
}

```

```

void iniciar_lecturas(){
    unsigned long now = millis();
    if (now - lastMsg > 1000) { // tiempo de muestra
        lastMsg = now;
        aceleraciones_amplitudes();
        frecuenciax();
        frecuenciay();
        frecuenciaz();
        String dato_JSON;
        DynamicJsonDocument doc(JSON_OBJECT_SIZE(11)); // creamos el documento
y definimos su tamaño
        JsonObject obj = doc.createNestedObject("Object"); // creamos el objeto
        obj["Acx"]=acx;
        obj["Acy"]=acy;
        obj["Acz"]=acz;
        obj["Ampx"]=sx;
        obj["Ampy"]=sy;
        obj["Ampz"]=sz;
        obj["Frz"]=fz;
        obj["Frx"]=fx;
        obj["Fry"]=fy;
        obj["Frz"]=fz;
    }
}

```

```

    obj["Pacienteid"]=count;
    DynamicJsonDocument doc2(JSON_OBJECT_SIZE(14)); // creamos el
documento y definimos su tama;o
    doc2["obj"]=obj;
    serializeJson(doc2, dato_JSON); //funcion para convertir el Objeto JSON a String,
aqui ocupamos la varibale dato_JSON Serial.println(obj);
    Serial.println(dato_JSON);
    client.publish("datos/monitoreo",String(dato_JSON).c_str(),true);//topic, dato que
va air aumentando en 1
}
}

```

```

void loop(){
    byte buttonState = digitalRead(button1);
    byte buttonState1 = digitalRead(button2);
    byte buttonState2 = digitalRead(button3);
    //Adicinar 1
    if (buttonState == LOW) {
        count=count+1;
        Serial.print("Aumentar: ");
        Serial.println(count);
        String dato_JSON;
        DynamicJsonDocument doc(JSON_OBJECT_SIZE(10)); // creamos el documento
y definimos su tamaño
        JsonObject obj = doc.createNestedObject("Object"); // creamos el objeto
        obj["Pacienteid"]=count;
        DynamicJsonDocument doc2(JSON_OBJECT_SIZE(13)); // creamos el
documento y definimos su tama;o
        doc2["obj"]=obj;
        serializeJson(doc2, dato_JSON); //funcion para convertir el Objeto JSON a String,
aqui ocupamos la varibale dato_JSON Serial.println(obj);
        Serial.println(dato_JSON);
    }
}

```

```

    client.publish("datos/paciente",String(dato_JSON).c_str(),true);//topic, dato que va
air aumentando en 1
}
//Disminuir 1
if (buttonState1 == LOW) {
    if (count<1){
        count=1;
        Serial.print("Error: ");
        Serial.println(count);
        String dato_JSON;
        DynamicJsonDocument  doc(JSON_OBJECT_SIZE(10)); // creamos el
documento y definimos su tamaño
        JsonObject obj = doc.createNestedObject("Object"); // creamos el objeto
        obj["Pacienteid"]=count;
        DynamicJsonDocument  doc2(JSON_OBJECT_SIZE(13)); // creamos el
documento y definimos su tama;o
        doc2["obj"]=obj;
        serializeJson(doc2, dato_JSON); //funcion para convertir el Objeto JSON a String,
aquí ocupamos la variable dato_JSON  Serial.println(obj);
        Serial.println(dato_JSON);
        client.publish("datos/paciente",String(dato_JSON).c_str(),true);//topic, dato que
va air aumentando en 1
    }
    else {
        count=count-1;
        Serial.print("Disminuir: ");
        Serial.println(count);
        String dato_JSON;
        DynamicJsonDocument  doc(JSON_OBJECT_SIZE(10)); // creamos el
documento y definimos su tamaño
        JsonObject obj = doc.createNestedObject("Object"); // creamos el objeto
        obj["Pacienteid"]=count;

```

```

    DynamicJsonDocument  doc2(JSON_OBJECT_SIZE(13)); // creamos el
documento y definimos su tama;o
    doc2["obj"]=obj;
    serializeJson(doc2, dato_JSON); //funcion para convertir el Objeto JSON a String,
aquí ocupamos la varibale dato_JSON Serial.println(obj);
    Serial.println(dato_JSON);
    client.publish("datos/paciente",String(dato_JSON).c_str(),true);//topic, dato que
va air aumentando en 1

    }
}

//Iniciar
if (buttonState2 == LOW) {
    iniciar_lecturas();
}
delay(100);

if (!client.connected()) {
    reconnect();
}
client.loop();
}

```

### **Anexo 7. Código de Arduino dispositivo de terapia**

```

#include <WiFiManager.h>
#include <PubSubClient.h>
#include <Arduino.h>
#include <ArduinoJson.h>
#include <ezButton.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

```

```

LiquidCrystal_I2C lcd(0x27,16,2);
//Pines botones
#define button1 5 //D1
#define button2 4 //D2
//#define button3 0 //D3
int count=0; //contador

WiFiClient espClient;
PubSubClient client(espClient);
//Dirección IP del broker MQTT para conectarse a este
const char* mqtt_server = "20.38.46.67";

//Pines Inicio
ezButton limitSwitch(10); //Pin SD3
ezButton limitSwitch1(2); //Pin D4
ezButton limitSwitch2(0); //Pin D3
//Pines Final
ezButton limitSwitch3(14); //Pin D5
ezButton limitSwitch4(12); //Pin D6
ezButton limitSwitch5(13); //Pin D7
// Cronómetro
unsigned long inicio, inicio1, inicio2, finalizado, finalizado1, finalizado2;
unsigned long Ttranscurrido, Ttranscurrido1, Ttranscurrido2, TtranscurridoT;

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Create a random client ID
    String clientId = "ESP8266Client-Cronometro";
    clientId += String(random(0xffff), HEX);
    // Attempt to connect

```

```

if (client.connect(clientId.c_str())) {
    Serial.println("connected");
    // Once connected, publish an announcement...
    client.publish("datos/tiempo", "Conectado");
    // ... and resubscribe
    client.subscribe("datos/#");
} else {
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    // Wait 5 seconds before retrying
    delay(5000);
}
}
}

void setup(){lcd.backlight();
    Serial.begin(115200);
    lcd.begin(16, 2);
    lcd.init();
    lcd.backlight();

    lcd.setCursor(0, 0);
    lcd.print("Bienvenido");

    client.setServer(mqtt_server, 1883);
    WiFiManager wifiManager; // Creamos una instancia de la clase WiFiManager
    //wifiManager.resetSettings(); // Descomentar para resetear configuración
    wifiManager.autoConnect("ESP8266-Cronometro"); // Creamos AP y portal cautivo
    Serial.println("¡Conectado!");
    limitSwitch.setDebounceTime(50);
    limitSwitch1.setDebounceTime(50);
    limitSwitch2.setDebounceTime(50);

```

```

limitSwitch3.setDebounceTime(50);
limitSwitch4.setDebounceTime(50);
limitSwitch5.setDebounceTime(50);
//iniciar botones en alto
pinMode(button1,INPUT_PULLUP);
pinMode(button2,INPUT_PULLUP);
//pinMode(button3,INPUT_PULLUP);
}

void MostrarResultado(){
    float h, m, s, ms;
    unsigned long terminado;
    Ttranscurrido = finalizado - inicio;
    Ttranscurrido1 = finalizado1 - inicio1;
    Ttranscurrido2 = finalizado2 - inicio2;
    TtranscurridoT = Ttranscurrido + Ttranscurrido1 + Ttranscurrido2;
    h = int(TtranscurridoT / 3600000);
    terminado = TtranscurridoT % 3600000;
    m = int(terminado / 60000);
    terminado = terminado % 60000;
    s = int(terminado / 1000);
    ms = terminado % 1000;
    Serial.print("Tiempo transcurrido : ");
    Serial.println(TtranscurridoT);
    Serial.print("Tiempo transcurrido detallado: ");
    Serial.print(h, 0);
    Serial.print("h ");
    Serial.print(m, 0);
    Serial.print("m ");
    Serial.print(s, 0);
    Serial.print("s ");
    Serial.print(ms, 0);
    Serial.println("ms");
}

```

```

Serial.println();
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Tiempo:");
lcd.setCursor(0,1);
lcd.print("m:");
lcd.print(m,0);
lcd.print(" s:");
lcd.print(s,0);
lcd.print(" ms:");
lcd.print(ms,0);

String dato_JSON;
DynamicJsonDocument doc(JSON_OBJECT_SIZE(10)); // creamos el documento
y definimos su tamaño
JsonObject obj = doc.createNestedObject("Object"); // creamos el objeto
obj["Minutos"]=m;
obj["Segundos"]=s;
obj["Milisegundos"]=ms;
obj["Pacienteid"]=count;
DynamicJsonDocument doc2(JSON_OBJECT_SIZE(13)); // creamos el
documento y definimos su tama;o
doc2["obj"]=obj;
serializeJson(doc2, dato_JSON); //funcion para convertir el Objeto JSON a String,
aqui ocupamos la varibale dato_JSON Serial.println(obj);
Serial.println(dato_JSON);
client.publish("datos/tiempo",String(dato_JSON).c_str(),true);//topic, dato que va
air aumentando en 1
}

void loop(){
//Primero se llama para ejecutar la funcion loop

```



```

limitSwitch.loop();
limitSwitch1.loop();
limitSwitch2.loop();
limitSwitch3.loop();
limitSwitch4.loop();
limitSwitch5.loop();

byte buttonState = digitalRead(button1);
byte buttonState1 = digitalRead(button2);
// byte buttonState2 = digitalRead(button3);
//Adicinar 1
if (buttonState == LOW) {
    count=count+1;
    Serial.print("Aumentar: ");
    Serial.println(count);
    String dato_JSON;
    DynamicJsonDocument doc(JSON_OBJECT_SIZE(10)); // creamos el documento
y definimos su tamaño
    JsonObject obj = doc.createNestedObject("Object"); // creamos el objeto
    obj["Pacienteid"]=count;
    DynamicJsonDocument doc2(JSON_OBJECT_SIZE(13)); // creamos el
documento y definimos su tama;o
    doc2["obj"]=obj;
    serializeJson(doc2, dato_JSON); //funcion para convertir el Objeto JSON a String,
aquí ocupamos la varibale dato_JSON Serial.println(obj);
    Serial.println(dato_JSON);
    client.publish("datos/pacienteid",String(dato_JSON).c_str(),true);//topic, dato que
va air aumentando en 1

}

//Disminuir 1
if (buttonState1 == LOW) {

```

```

if (count<1){
    count=1;
    Serial.print("Error: ");
    Serial.println(count);
    String dato_JSON;
    DynamicJsonDocument  doc(JSON_OBJECT_SIZE(10)); // creamos el
documento y definimos su tamaño
    JsonObject obj = doc.createNestedObject("Object"); // creamos el objeto
obj["Pacienteid"]=count;
    DynamicJsonDocument  doc2(JSON_OBJECT_SIZE(13)); // creamos el
documento y definimos su tama;o
    doc2["obj"]=obj;
    serializeJson(doc2, dato_JSON); //funcion para convertir el Objeto JSON a String,
aqui ocupamos la varibale dato_JSON  Serial.println(obj);
    Serial.println(dato_JSON);
    client.publish("datos/pacienteid",String(dato_JSON).c_str(),true);//topic, dato que
va air aumentando en 1
}
else {
    count=count-1;
    Serial.print("Disminuir: ");
    Serial.println(count);
    String dato_JSON;
    DynamicJsonDocument  doc(JSON_OBJECT_SIZE(10)); // creamos el
documento y definimos su tamaño
    JsonObject obj = doc.createNestedObject("Object"); // creamos el objeto
obj["Pacienteid"]=count;
    DynamicJsonDocument  doc2(JSON_OBJECT_SIZE(13)); // creamos el
documento y definimos su tama;o
    doc2["obj"]=obj;
    serializeJson(doc2, dato_JSON); //funcion para convertir el Objeto JSON a String,
aqui ocupamos la varibale dato_JSON  Serial.println(obj);
    Serial.println(dato_JSON);

```

client.publish("datos/pacienteid",String(dato\_JSON).c\_str(),true);//topic, dato que va air aumentando en 1

```
    }  
}  
  
//Iniciar  
//if (buttonState2 == LOW) {  
//Finales de carrera de Inicio  
if(limitSwitch.isReleased()){  
    inicio = millis();  
    delay(300); // por rebote  
    Serial.println("Inicializar 1");  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print("Inicializar 1");  
    }  
if(limitSwitch1.isReleased()){  
    inicio1 = millis();  
    delay(300); // por rebote  
    Serial.println("Inicializar 2");  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print("Inicializar 2");  
    }  
if(limitSwitch2.isReleased()){  
    inicio2 = millis();  
    delay(300); // por rebote  
    Serial.println("Inicializar 3");  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print("Inicializar 3");  
    }  
}
```

```

//Finales de carrera final
if(limitSwitch3.isPressed()){
    finalizado = millis();
    delay(300);
    Serial.println("Fin 1");
    lcd.setCursor(0,1);
    lcd.print("Fin 1");
}
if(limitSwitch4.isPressed()){
    finalizado1 = millis();
    delay(300);
    Serial.println("Fin 2");
    lcd.setCursor(0,1);
    lcd.print("Fin 2");
}
if(limitSwitch5.isPressed()){
    finalizado2 = millis();
    delay(300);
    Serial.println("Fin 3");
    MostrarResultado();
}
delay(100);
if (!client.connected() {
    reconnect();}
client.loop();
}

```